Gnuplot を使ってグラフ作成

松尾行雄

ねらい

数値データをグラフに表示すると、そのデータの振る舞いを理解するのに役に立ちます。ここでは、グラフ作成をするアプリケーションの一つである「gnuplot」の操作を学習し体得することを目標とします。

最初に 2 次元プロットを例にとりながら基本操作を学習します。加えて 3 次元プロット、等高線プロットを表示する操作を学習します。

準備 1

terminal を開いて、ホームディレクトリ上に「GNU07」というディレクトリをつくり(mkdir GNU07)、以下の操作を行う。

cd GNU07
lftp 157.118.89.2
cd TG-Local/staff/matsuo
ls　（Gnuplot.pdf と stest2.dat があることを確認）
get Gnuplot.pdf
get stest2.dat
get sendai09.txt
exit

ファイルの中身

- Gnuplot.pdf：配布資料（pdf ファイルを開くには xpdf Gnuplot.pdf &を実行)。
- sendai09.txt：ある年の 9 月の日ごとの気温のデータが入っています。
- stest2.dat：３次元プロット時に使います。

０．GNUPLOT でできることを体験してみよう！

各自の NotePC に入っているデモ・ファイルを実行します。各自の NotcPC の「/usr/doc/gnuplot-4.2.0/demo」の中に「xxx.dem」というファイルがあります。これを全て自分の/home/GNU07/demo の中にコピーします。GNU07 にいることを確認して以下のように行いなさい。

mkdir demo
cd demo
cp /usr/doc/gnuplot-4.2.0/demo/* .　（←最後の「.」を忘れずに！）

(a-1) 以下のコマンドを実行して、デモを実行しなさい。

…$ gnuplot all.dem

注意：gnuplot を実行すると新しい画面が作られますが、ウインドのフォーカスを 1 度 terminal に戻す必要があります。terminal （gnuplot を起動した画面）をマウスでクリックしてフォーカスを戻しなさい。その後、gnuplot のメッセージを読んで行動しなさい。

(a-2)デモは、次のようなやり方でも実行できます。gnuplot を起動した後、以下のコマンドを実行しなさい（上記の注意を思い出しなさい)。地球儀の画面が出たら、マウスを使って地球儀を回転させて見なさい。

…$ gnuplot

gnuplot > load “world.dem”

(b) 今皆さんが居る demo ディレクトリには､様々なデモが保存されています。ファイルは、xxxxx.dem という形式の名前になっています。これらのデモファイルの中には、クジラが隠されていたりしますので、探してみましょう。

１． 基本操作

ここからは GNUPLOT の基本的な使い方を学びます。

(a) gnuplot の起動

terminal 上で「gnuplot」を実行します。

⇒「gnuplot>」の後に書かれている内容を入力して、実際にグラフを表示し、使い方を学んでください。

(b) 関数を 2 次元プロット（横軸を x 軸、縦軸を y 軸と呼びます）

関数を 2 次元プロットします。たとえば$y=2x^3+5x^2+x+1$ を表示させます。

gnuplot> plot 2*x**3+5*x**2+x+1

ここで「*」は掛け算のことで､「**」はベキ乗を表しています｡x の 3 乗は「x**3」とあらわします。x の範囲が-10 から 10 までのグラフが表示されます（このような x 軸と y 軸を用いた表示を 2 次元プロットと呼びます。また x 軸の範囲が初期設定となっています)。

次に三角関数 (sin, cos, tan 等があります) を表示させます。たとえば、y= sin(x) を表示させます。

gnuplot> plot sin(x)

x の値はラジアンで表示されています。ラジアンとは角度の単位で、2πラジアン=360（度）となります。ここでπは円周率(3.14159265…)となっています。

(c) 関数の複数表示

３つの関数を同時にプロットする。例えば、$y = \sin x$ と $y = \cos x$ と $y = \sin x / x$ の場合をやってみよう（コンマ(,)で区切って列挙すれば表示できます）
gnuplot> plot sin(x), cos(x), sin(x)/x

(d) 表示方法（補足説明）
データの表示方法は、with オプションによって変えることができます。
○with で指定できる代表的なオプションについて
points　　：各点の位置に小さなマーカを書きます（デフォルト）
lines　　：隣り合う点を直線で結びます
linespoints　：lines と points の両方を行います
impulses　　：各点から x 軸へ垂線を引きます
補足説明：「with lines」は「w l」に、「with points」は「w p」に、「with linespoints」は「w lp」に省略できます。
gnuplot> plot sin(x) with impulses
gnuplot> plot sin(x) with linespoints
→データファイルのプロットと同様に関数のプロットの場合も、GNUPLOT x(横軸)の範囲を何等分かして、その各分点の上での関数値（ここでは sin(x)の値）を計算し、それらを線分で結ぶ方法をとっていることがわかります。

(e) 関数プロットの場合のデータ点数（set samples）
gnuplot> plot sin(10*x)
上記のコマンドを実行したときにどのようにグラフが表示されるかを確認してください（関数を正しく表示できないことを確認)。
→これは描画したい関数のスケールに比べて、分点の数が少なすぎて粗いためです。デフォルトでは分点の数が 100 になっています。結果を正しく表示するためには十分な分点の数を用意する必要があります。そこで「set samples」を使って、分点の数を指定します。
gnuplot> set samples 1000
gnuplot> plot sin(10*x)

(f) 描画範囲（横軸の範囲を x 軸の範囲と縦軸の範囲を y 軸の範囲）の指定
◆x 軸の範囲と y 軸の範囲を指定してプロットできます。以下を入力して、x 軸と y 軸の範囲がどのように変化するかをみてみよう。
gnuplot> plot [-2*pi:2*pi] sin(x)
gnuplot> plot [-2*pi:2*pi] [-2:2] sin(x)
gnuplot> plot [][-2:2] sin(x)

gnuplot> plot [:2*pi][-2:] sin(x)
ここで、「pi」とは円周率πの値となります。

練習　x 軸の範囲が-10 から 10 で、y 軸の範囲が-4 から 4 で y=3*cos(x)+2*sin(x) のグラフを表示させなさい。

◆デフォルトのプロット範囲の指定（xrange,yrange,zrange）
gnuplot> set xrange [0:10]
gnuplot> plot sin(x)
gnuplot> plot cos(x)
gnuplot> reset
→set xrange (yrange, zrange) を使うと plot の度に範囲を書く必要がない。
　reset　: set した内容をリセット

(g) 関数の定義と軸ラベルの指定
　set xlabel, set ylabel, set zlabel で自動的に各軸がどのような量を表しているかを記入できます。
gnuplot> f(x) = 1/(1+exp(-x))
gnuplot> set xlabel 'x'
gnuplot> set ylabel 'f(x)'
gnuplot> plot f(x)
→「f(x)= 1 / (1+exp(-x))」で関数 f(x)の設定ができます。ここで「exp」は指数関数のことです。
gnuplot> reset
●xlabel と ylabel がグラフに表示されるかを確認して表記しなさい。
xlabel：（　　）
ylabel：（　　）

(h) 軸目盛りの指定（tics,xtics,ytics,ztics）
　set tics は軸の目盛の向きを内側にするか外側にするかを決定します。デフォルトは内向き(IN)です。外側に設定するときは set tics out と入力します。
gnuplot> set tics out
gnuplot> plot sin(x)
gnuplot> set tics
gnuplot> replot
確認　「set tics out」と「set tics」で目盛の向きが変わっていることを注目し

てください。

軸目盛の目盛幅を set xtics,set ytics, set ztics で設定します。
gnuplot> set xtics -9,3,9
gnuplot> plot sin(x)
→x の目盛を-9 から 9 まで 3 刻みで書いています。

練習　y 軸の目盛を-1 から 1 まで 0.25 刻みにして replot してください。

(i) x 軸または y 軸に点線を表示/非表示
x 軸（y=0）または y 軸(x=0)に点線を描画するかしないかを設定します。例えば x 軸に点線を表示するには set xzeroaxis と入力し、非表示にするには unset xzeroaxis と入力します。
gnuplot> set xzeroaxis
gnuplot> plot sin(x)
gnuplot> set yzeroaxis
gnuplot> replot
gnuplot> unset xzeroaxis
gnuplot> unset yzeroaxis
gnuplot> replot
gnuplot> reset

練習　各コマンドの意味をカッコ内に書いてください。
set yzeroaxis　　（　　）
unset yzeroaxis　（　　）
(j) 対数プロット(logscale)
実験などで得られたデータなどを対数目盛のグラフとして表示すると、データの変化の傾向をつかむのに大変便利です。set logscale は各軸の目盛を対数目盛に変換します。
gnuplot> set logscale y
gnuplot> plot exp(x)
gnuplot> unset logscale
gnuplot> replot
gnuplot> reset

(k) 余白の調整（offsets）

描画領域の余白は set offsets で行います。

gnuplot> set offsets 1,1,1,1

gnuplot> plot sin(x)

→描画領域の左右上下の余白を一目盛分増やして表示する。数字の順序は左、右、上、下である。値を変えて余白の変化を確認してください。

gnuplot> reset

(l) 描画時刻の表示(time)と背景を方眼(grid)にする

set time で描画時刻を表示します。

gnuplot> set time

gnuplot> plot sin(x)

gnuplot> unset time

gnuplot> replot

描画下地に方眼を描きます。

gnuplot> set grid

gnuplot> plot sin(x)

gnuplot> unset grid

gnuplot> replot

(m)　コメント、凡例などの記入

◆ 凡例の非表示(unset key)

gnuplot> plot sin(x)

gnuplot> unset key

gnuplot> replot

gnuplot> set key default

gnuplot> replot

gnuplot> reset

練習　unset key によって、グラフの右上にある凡例（sin(x)）がどのように変化したかを確認してカッコ内に書きなさい

（　　）

◆ グラフの任意の場所にコメントを挿入(set label)

gnuplot> set label 'x=0,y=0' at 0,0

gnuplot> plot sin(x)

→位置(0,0)に「x=0,y=0」というコメントを挿入

gnuplot> set label ‘x=3,y=0’ at 3,0 center
gnuplot> replot
→位置(3,0)に「x=3,y=0」というコメントを中央揃えで挿入（デフォルトは左寄せになっています)。
→一旦表示したラベルを消したり、内容を変更したいする場合はラベルのタグ番号を用いて行います。ラベルは複数個表示できますから、GNUPLOT では表示したラベルにタグ番号をつけてそれらを管理しています。どのラベルがどのタグ番号に対応しているかを見る場合は show label と入力します。
gnuplot> show label
→設定した順番にラベルにタグがついていることがわかります。タグ番号がわかれば、ラベル内容の変更はこの番号を使って実行できます。
gnuplot> set label 1 at 2,0.5
gnuplot> replot
→ラベル 1 の場所を(2,0.5)に変更
gnuplot> set label 2 right
gnuplot> replot
→ラベル 2 を右寄せに変更
gnuplot> unset label 1
gnuplot> replot
→ラベル 1 を消去
gnuplot> reset

◆ 矢印の挿入(arrow)
　ある特定の箇所などを指示するために矢印を書くこともできます。そのために set arrow を使います。
gnuplot> set xrange[-3:3]
gnuplot> set yrange[-14:14]
gnuplot> plot –x**2+2*x+1 w l
始点の座標が（2,10）で、終点の座標が(1,2)の矢印を挿入し、始点のところに「Maximal」というコメントを挿入します。
gnuplot> set arrow from 2,10 to 1,2
gnuplot> set label ‘Maximal’ at 2,10
gnuplot>replot
→set arrow も set label と同様にタグ番号で管理されており、その内容の変更そして削除はタグ番号を介して行います。
gnuplot> show arrow

→矢印のタグ番号を確認
gnuplot> set arrow 1 to -1,-2
gnuplot> replot
gnuplot> unset arrow 1
gnuplot>replot

2. 設定した内容の保存と利用

「set」を使って設定した項目（set arrow とか set xrange など）をファイルに保存し、利用することができます。

・ save ‘ファイル名.gnu’ :「set」で設定した内容をファイルに保存します。
・ load ‘ファイル名.gnu’ : 保存した内容を入力します。

ここでは「矢印の挿入」のところで設定した内容をファイル（ファイル名：arrow.gnu）に保存し、リセット後、ファイルを読み出し、グラフを出力させます。

gnuplot> set arrow from 2,10 to 1,2
gnuplot> save ‘arrow.gnu’
gnuplot> replot （保存した内容の確認）
gnuplot> reset (リセット)
gnuplot> replot （矢印がないことを確認）
gnuplot> load ‘arrow.gnu’
gnuplot> reset

3. データファイルのグラフ表示

数値計算や測定などで得られたデータを表示する方法について述べます。そのために GNUPLOT が読み込むことができるデータを用意する必要があります。データファイルに関する注意事項は次のとおりです。

（１）データ間はスペースもしくはタブで区切られていること。
（２）#はコメントを表す
（３）データの空行はデータのつながりをきります。

ここでは、ある年の 9 月の仙台の気温データ（日ごとの平均、最高、最低気温）である「sendai09.txt」というファイルを用います。「sendai09.txt」のファイルの中身を確認してください(mule 等を使って)。

このファイルの最初の行は#が行頭に書いてあるのでコメントになります。他の行に関しては、1 列目から 4 列目で、日、平均気温、最高気温、最低気温をあらわしています。

```
#   日   平均気温   最高気温   最低気温
1        22.7    26.1    19.4
2        22.3    26.9    17.9
3        23.8    29      19.9
4        23.5    27.4    19.4
・・・・・
```

練習　以下のように実行したときに表示されるグラフの横軸と縦軸は何を表しているかをカッコ内に書きなさい。

gnuplot> set style data linespoints
gnuplot> plot ‘sendai09.txt’
横軸　（　　）　縦軸　（　　）

・set style data：予めデータの表示方法を設定する方法です。data の後に表示方法を書きます。線の場合は「lines」、ポイント(小さなマーカー)の場合は「points」、線とポイントの両方を描画する場合は「linespoints」と書きます。

●「sendai09.txt」は 4 列のデータです。グラフに表示させる縦軸と横軸を指定するには「using」を使います。例えば、一列目（日）を横軸に、３列目（最高気温）を縦軸に表示させるには次のように実行します。

gnuplot>plot ‘sendai09.txt’ using 1:3

グラフの右上の凡例には「sendai09.txt　using 1:3」と表示されています。この凡例を「minimum temperature」と表示させるには次のようにします。

gnuplot>plot ‘sendai09.txt’ using 1:3 title ‘minimum temperature’

●複数のデータを同時に表示させることもできます。例えば、「平均気温」と「最高気温」のデータを一つのグラフに表示させるにはカンマで区切って、データを指定します（横軸は日です)。

gnuplot>plot ‘sendai09.txt’ using 1:2, ‘sendai09.txt’ using 1:3

練習　「平均気温」、「最高気温」、「最低気温」のデータを一つのグラフに表示させなさい。

4. ファイル出力

グラフを画面に表示させるだけではなく、ファイルに保存することができます。たとえば、$y=x^3+2x^2+1$ のグラフをps(ポストスクリプト)ファイル（ファイル名：'test.ps'）に出力するには以下の操作を行います。

```
gnuplot> plot x**3+2*x**2+1
gnuplot> set term postscript          (1)
gnuplot> set output 'test.ps'         (2)
gnuplot> replot                       (3)
gnuplot> set output
```

(1) ファイルの種類の設定

・eps ファイルに出力する場合は「set term postscript eps」に変更

・png ファイルに出力する場合は「set term png」に変更

(2) ファイル名を設定（set output 'ファイル名'）

拡張子はそれぞれ ps,eps,png とする。

(3) プロット(plot ~, splot ~, replot)

(4)プロットが終了したら、最後に「set output」を入力してください。

●ファイルに保存作業が終了して、再び画面への出力にしたい場合は次のように入力してください。確認のため、y=cos(x)のグラフを表示させます。

```
gnuplot> set terminal x11
gnuplot> plot cos(x)
```

●出力したファイルを表示

出力した ps ファイルを表示するには以下のように入力します。

```
gnuplot> !gv 'test.ps'
```

gnuplot を一旦終了（exit）

terminal 上で ps ファイルを表示するには

```
gv 'test.ps'
```

と入力します。

５．3 次元プロット

terminal 上で「gnuplot」を起動する。

(a) 一次元データの 3 次元プロット

新しいデータファイル「stest.dat」を以下の内容で mule を使って作成する。

```
#stest.data
0.5
0.3
0.2
0.4
0.8
```

保存後終了し、gnuplot を起動する。

gnuplot> set xlabel ‘x’

gnuplot> set ylabel ‘y’

gnuplot> set zlabel ‘z’

gnuplot> splot ‘stest.dat’ w l

→3 次元空間上に線が描かれます。この時 GNUPLOT はこれらのデータを自動的に(x, y, z) = (0, 0, 0.5), (1, 0, 0.3), (2, 0, 0.2), (3, 0, 0.4), (4, 0, 0.8)と解釈してそれを 3 次元空間上にプロットします。

練習　視点の変更をして、プロットされた座標の位置を確認しよう。

◆ 視点とは

3 次元で描写されたグラフは見た角度（視点の位置）により変化する。この視点の位置を変えることによって、グラフは見やすくなったり、見にくくなったりします。

グラフの左下にある「view: 60.0000, 30.0000」が視点の角度を表しています。デフォルト（初期設定）では 60 度と 30 度となっており、「x 軸周りに 60 度回転し、z 軸周りに 30 度回転する」という意味となります。これらの角度を変更することによって、視点の変更を行います。

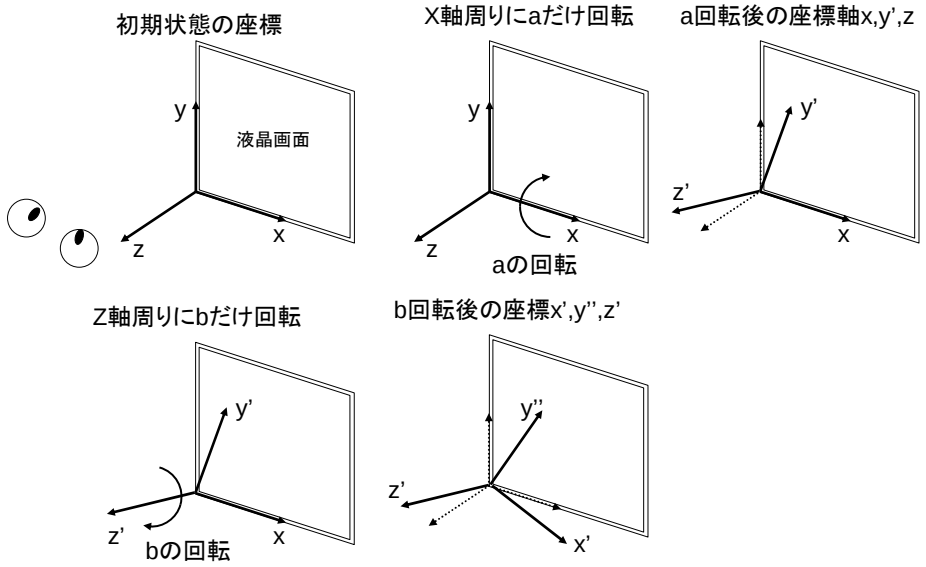
◆ 視点変更の方法

（１）描写されたグラフ上でマウスの左ボタンを押しながら左右上下の移動を行い、視点の変更を行う。

（２）描写されたグラフを選択し、キーボード上にある矢印キー（左・右・上・下）を押して、視点の変更を行う。

（３）set view を用いて視点の変更を行う。

⇒「視点変更の方法」の（1）または（2）を使って、回転角度を変更することにより、プロットされた座標の位置が確認しやすくなります。(x 軸周りの回転角度、z 軸周りの回転角度）を(90 度、0 度)そして（0 度、0 度）にしたときの座標の位置を見てみよう。



上図を使って回転について説明する。たとえば、x 軸周りの回転角度と z 軸まわりの回転角度が 0 度の場合、グラフが描かれる座標系はモニタの左右方向に x 軸、モニターの上下方向に y 軸、モニタに立てた垂線の方向が z 軸になります。x 軸のまわりの回転角度（0 から 180 度）を a とし、z 軸まわりの回転角度(0 から 360 度)を b としたときの視点の変化についてみていきます。最初に x 軸周りに a だけ回転させ（y 軸→y'軸、z 軸→z'軸)、次に z'軸周りに b だけ回転させます（y'軸→y"軸、x 軸→x'軸)。

◆図のスケールの変更

3 次元で描写されたグラフは図の拡大率（スケール）も設定することができます。グラフの左下にある「scale: 1.00000, 1.00000」が拡大率を表しています。２つの数字のうち、最初の数字がグラフの拡大率を、次の数字が z 軸の拡大率を表しています。

◆set view を用いて、視点と拡大率を変更する

たとえば、x 軸周りの回転角度を 45 度、z 軸周りの回転角度を 120 度、グラフの拡大率を 1 倍、z 軸方向の拡大率を 0.75 倍にするときは次のように設定し

ます。
gnuplot>set view 45,120,1,0.75
gnuplot>replot

◆データの空行について
stest.dat の中身を以下のように変更後保する。0.3 と 0.2 の行の間に空行を挿入する

```
#stest.data
0.5
0.3

0.2
0.4
0.8
```

gnuplot>set view 60,30,1,1
gnuplot> splot 'stest.dat' w l

練習　プロットされた座標の位置を確認して書いてください。回転角度を変えてみましょう。(x の値, y の値, z の値)のように書いてください
(　,　,　),(　,　,　),(　,　,　),(　,　,　),(　,　,　)

→stest.dat の中身を次のように変更して splot する。

```
#stest.data
0.5
0.3

0.2
0.4

0.8
0.3
```

gnuplot> set view 60,30,1,1
gnuplot> splot 'stest.dat' w l
→空行は、データ系列を y 方向へずらす働きとして解釈されている。
→ (0, 0, 0.5), (1, 0, 0.3)と(0, 1, 0.2), (1, 1, 0.4)と (0, 2, 0.8), (1,2, 0.3)のように解釈して、y 座標の等しいデータどうしを線分でプロットしています。さらに空行で区切られている連続するデータの個数が相等しいときは、x 座標の等しいどうしも線分で結んでいます。

◆ 関数プロット

$z=\exp(-x)(1-y^2)$をプロットする。

gnuplot> splot exp(-x)*(1-y*y)

→範囲指定を行う。x,y,z それぞれの範囲を 0 から 1 とする

gnuplot> splot [0:1][0:1][0:1] exp(-x)*(1-y*y)

または

gnuplot> set xrange[0:1]

gnuplot> set yrange[0:1]

gnuplot> set zrange[0:1]

gnuplot> splot exp(-x)*(1-y*y)

gnuplot> reset

(b) 3 次元データの 3 次元プロット

新しいデータファイル「stest1.dat」を以下の内容で mule を使って作成する。

```
0.0 0.0 0.2
0.1 0.2 0.9
0.2 0.4 1.4
0.3 0.6 0.8
0.1 -0.2 0.9
0.2 -0.4 0.1
0.3 -0.6 0.3
```

一列目が x の値、二列目が y の値、３列目が z の値に対応するように 3 次元プロットする。set parametric でそれぞれの列が x,y,z の値であることを指定する。

gnuplot> set parametric

gnuplot> set xlabel ‘x’

gnuplot> set ylabel ‘y’

gnuplot> set zlabel ‘z’

ラベルの指定を一行で次のように書くこともできます。

gnuplot> set xlabel ‘x’; set ylabel ‘y’; set zlabel ‘z’

→セミコロン(;)を使うことで複数の指定がかけます。

gnuplot> splot ‘stest1.dat’ w l

◆ 空行で区切られた 3 次元データの 3 次元プロット

最初に stest2.dat の中身を確認する(more stest2.dat 等を使って)。21 個ごとに空行があることを確認し、GNUPLOT 上で 3 次元プロットする。

gnuplot> splot ‘stest2.dat’
gnuplot> splot ‘stest2.dat’ w l

stest2.dat の関数は

$$z = \exp\left(-\frac{x^2 + y^2}{40}\right)$$

です。一列目が x に、二列目が y に、三列目が z に対応します。x の値の範囲は -10 から 10 で、y の値の範囲は-10 から 10 です。

練習　上記の 3 次元プロットした図を「curve3d.ps」という名前の ps ファイルに保存しなさい（9 ページを参照してください)。

(c) 等高線プロット
3 次元の面が与えられたときにその等高線を同時に書きたいことがよくあります。その設定をしてくれるのが set contour です。
stest2.dat を使って等高線プロットを行います。
gnuplot> set contour
gnuplot> splot ‘stest2.dat’ w l

◆set contour では base（デフォルト）,surface, both の 3 つのオプションが指定できます。base はプロットの底面に等高線を書きます。surface は面の上に、both は base と surface の両方に等高線を書きます。
gnuplot> set contour surface
gnuplot> replot
gnuplot> set contour both
gnuplot> replot

◆ 等高線の間隔を指定する（set cntrparam levels increment)
gnuplot> set cntrparam levels increment 0,0.15,1
→開始値 0 で終了値 1 の間を 0,15 ずつ増やして等高線を引くように指定します。
gnuplot> replot
→set cntrparam levels increment a,b,c(上の例では a が 0 、b が 0.15、c が 1)のように 3 つの数字を入力することで等高線の開始値(a)と終了値(c)の間の等間隔の値(b)に対する等高線を引くことができます。

練習　開始値0で終了値1の間を0,1ずつ増やして等高線を引くように指定し、グラフを表示させなさい。

(d) 等高線だけのプロット

　等高線だけをプロットするには、等高線の表示の設定と局面の非表示の設定と視点の変換が必要となります。

gnuplot> set contour
gnuplot> replot
gnuplot>unset surface
gnuplot> set view 0,0,1,1
gnuplot> replot
gnuplot> reset

練習　次のコマンドの意味をカッコ内に書きなさい。
unset surface：（　　）

●gnuplot 上で cd,ls 等の操作を行うには
・現在のディレクトリを確認する
gnuplot>pwd
・ディレクトリを移動する
例　GNU06 というディレクトリに移動する
gnuplot>cd ‘GNU06’
・ディレクトリのファイルを表示する
gnuplot>!ls

６．gnuplot の参考のページ

　以下のホームページは GNUPLOT の参考ページですので一読してください。例えば、gnuplot でどのようなことができるのかを調べて、自分で試してみなさい（本日の講義で時間が余ってしまった場合は、良い機会なので必ず読みなさい)。

http://t16web.lanl.gov/Kawano/gnuplot/
http://gnuplot.sourceforge.net/demo/

GNUPLOT の応用編：媒介変数表示について

媒介変数表示について

GNUPLOT で２次元関数を表示するときは、y=f(x)という形で表しますが、媒介変数を用いて

x = f(t)

y = g(t)

のように表示することもできます（媒介変数表示)。この形式を用いることで、複雑な関数を GNUPLOT に表示させることが可能になります。

媒介変数を用いてグラフを描くには、まずset parametricを用いて、関数が媒介変数表示であることを明示します。plotに続けて、x座標となる関数f(t)とy座標のg(t)をplot f(t),g(t)のように与えます。例えばf(t)=t, g(t)=t^2の場合は次のように書きます。

```
gnuplot> set parametric
gnuplot> plot t,t**2
```

●垂直線の書き方

最も単純かつ y=f(x)という関数型で表現できないのが、x=const(定数、例えばx=3)という垂直の線です。この縦線は媒介変数 t を用いると、x = const, y=t と書き、t を適当な範囲で変化させた場合に相当します。t の範囲は set trange で設定できます。

```
gunplot> set parametric
gnuplot> const=3
gnuplot> set trange[1:4]
gnuplot> set xrange[0:5]
gnuplot> set yrange[0:5]
gnuplot> plot const,t
gnuplot>set trange[0:5]
gnuplot> replot
```

→縦線が引かれます。const に 3 という値を代入しています。

```
gnuplot> reset
```

●円の書き方

円の媒介変数表示は、x=cos(t), y=sin(t) で与えられ、t を 0 から 2π まで変化させれば円が描けます。円をきれいに描くために set size というコマンドも用いています。

gnuplot> set parametric
gnuplot> set size square
→正方形のグラフを指定します。
gnuplot> set xrange[-1:1]
gnuplot> set yrange[-1:1]
gnuplot> plot [0:2*pi] cos(t), sin(t) title 'circle'
→円が描かれます。
ちょっとした追加
(1)正多角形の表示
gnuplot> set samples 8
gnuplot> replot
→正 7 角形が表示されます。set samples によってグラフに表示させる分点の数を決めています。この場合、0 から 2π の範囲で 8 個の点のみを用いてグラフを表示しています。
(2)渦巻き
x=t cos(t), y=t sin(t)として円の半径が t に依存して変化させてプロットする。
gnuplot> set samples 200
→分点の数を 200 にする
gnuplot> set xrange[-10*pi:10*pi]
gnuplot> set yrange[-10*pi:10*pi]
gnuplot> plot [0:10*pi] t*cos(t),t*sin(t)

●x 軸と y 軸を入れ替える
　関数表示は普通 y=f(x)と書かれますが、媒介変数を用いると x=f(y)のグラフを表示させることができます。例えば x=f(t)=2πsin(y)と y=2πcos(x)を表示させます。
gnuplot> set parametric
gnuplot> set size square
gnuplot> set trange[-2*pi:2*pi]
gnuplot> set xrange[-2*pi:2*pi]
gnuplot> set yrange[-2*pi:2*pi]
gnuplot> plot 2*pi*sin(t), t with lines, t, 2*pi*cos(t) with impulses