5 グラフ作成 - GNUPLOT -

GNUPLOT は基本的には関数をグラフ化して表示するツールです。出力形式が豊富で各種のプリンター言語を理解します。EPS はもちろんのこと、嬉しいことに tgif の .obj もサポートしています。したがって、後で tgif の上で加工して見栄えの良い図を作成可能となります。また、点列のプロットもできますから、実験データを読み込んでグラフを作成する目的にも使えます。さらに言えば、スクリプトで制御できるので定型作業の自動化も可能と、まあ至極便利なツールなのです。

5.1 対話的な使い方

対話モードで使ってみましょう。ただ単に

```
gnuplot
```

と入力します。すると、GNUPLOT がプロンプト "gnuplot>" を表示してコマンド入力を促してきます。

```
      G N U P L O T
      Linux 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   Colin Kelley, Thomas Williams

      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'
gnuplot>
```

最後の行の 1 つ前の行ではターミナルが'x11' とありますから、図が X に出力されます。

5.1.1 2 次元プロット - plot -

そこで、1 年生の物理実験でお馴染みのフラウンフォーファー回折強度の理論曲線

$$I(x) = \left(\frac{\sin x}{x}\right)^2 I_0$$

の図を描かせましょう。

```
gnuplot> plot (sin(x)/x)**2
```

と入力してください。別の窓が現れてグラフが表示されます (図 29 参照)。x 軸の範囲はデフォフトでは -10 から 10 です。軸の目盛 (**tics** と呼ばれる) も適当につけられます。もちろん範囲や目盛や軸の名前を任意に設定することもできます (後述)。y 軸の範囲はデフォルトでは全てが図面内に収まるように自動調整されます。

5.1.2 EPS ファイルの出力 - set term *terminal* と set output *'filename'* -

では、この図を EPS 形式でファイルに出力しましょう。

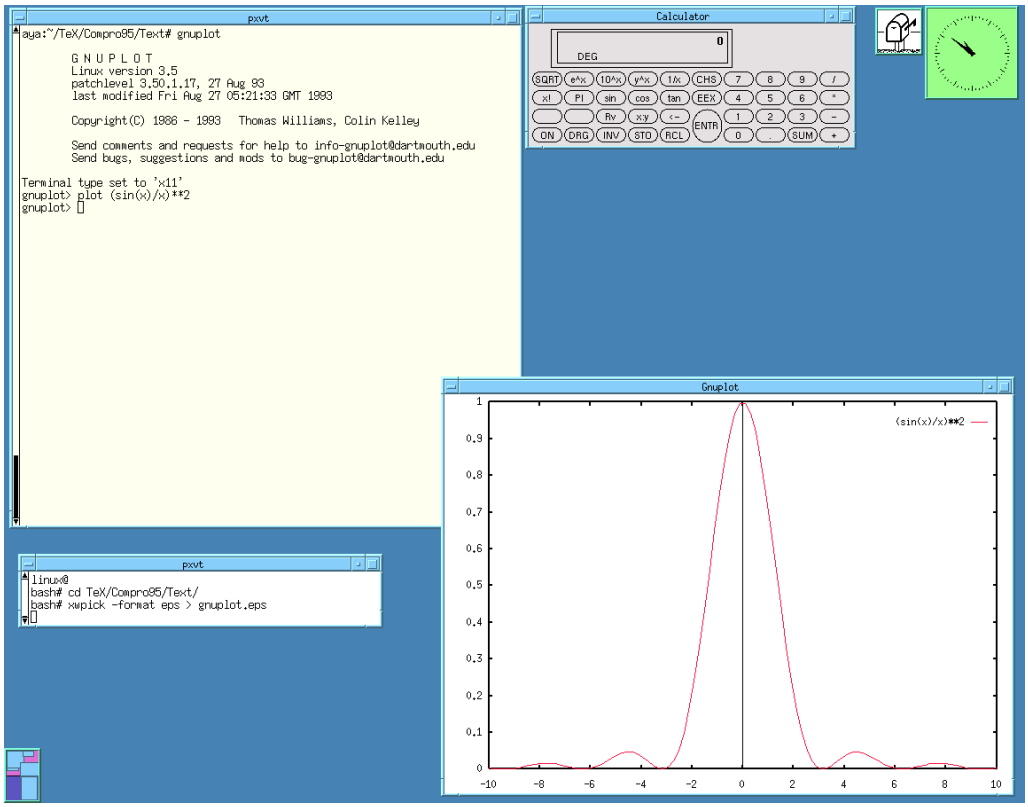

図 29　gnuplot は X では別窓にグラフを表示する

EPS 形式で出力する方法

```
gnuplot> set term postscript eps 22
gnuplot> set output 'sinxx.eps'
gnuplot> replot
```

"*set term ...*" によりターミナル形式を EPS に変えます。"*set output 'filename.eps'*" で出力先をファイル *filename.eps* にします。そして、同じプロットを再度出力するのは "*replot*" で行います。

5.1.3　シェルでのコマンド実行 - !arg -

ファイルができているか確認しましょう。

```
gnuplot> !ls sinxx.eps -lt
```

"! (エクスクラメイションマーク)" の後のコマンドはシェルに実行させます。従って、sinxx.eps を PS(EPS) 図形のビューア **ghostscript** で見たい場合は、

```
gnuplot> !gs sinxx.eps
```

で可能です。

> UNIX の標準コマンドについては,"コンピュータ実習入門"の第 18 章を参考にしてください。ファイルに関するコマンド **ls, cat, less, cp, rm, mv** は最低限必要です。ただし、細かいオプションが SUN と Linux(GNU) で異なっている場合もあるので、**man** コマンドで確かめる習慣を見につけたいものです。

この EPS ファイルを **epsbox.sty** で取り込んだのが次の図 **30** です。

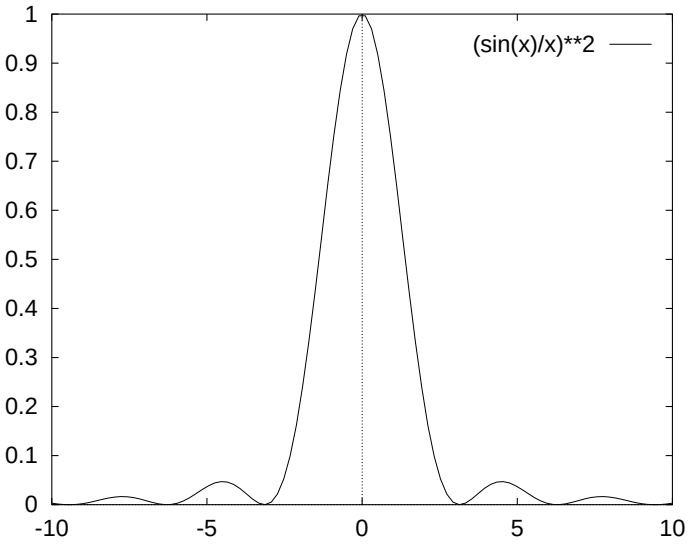

図 **30** **EPS** 形式で保存したものを取り込んだ場合

5.1.4 終了 - quit -

GNUPLOT を終了するコマンドは **quit** です。

5.1.5 データ (点列) の表示 - plot '*filename*' [with *style*] -

次のような内容のデータファイル 'sqx.dat' があったとします。

```
0   0    0
0.5 0.25 0.125
1.0 1.00 1.000
1.5 2.25 3.375
 ...
 ...
5.0 25.00 125.000
5.5 30.25 166.375
```

これを GNUPLOT で表示してみましょう。対話モードで起動して、plot を実行します。

```
gnuplot> plot 'sqx.dat'
```

何もオプション指定しなければデータファイルの第 1 列を x の値、 第 2 列を y の値として、マーク (**point**) で描きます。上のプロットに重ねて、第 1 列と第 3 列を x, y の値とした折れ線 (**line**) を描くには次のように ',' で全体を区切って並べ列指定に **using i:j** を用いればいいのです。

```
gnuplot> plot 'sqx.dat' title 'x**2', 'sqx.dat' using 1:3 title 'x**3' with line
```

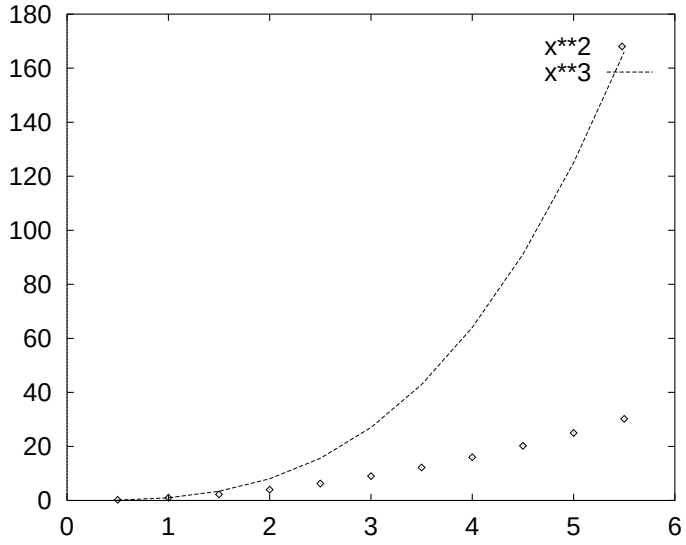

図 31　データ (点列) ファイルのグラフ化の例

ちょっと寄り道

ところで、このテストのためのデータファイルはどのようにして作成したらいいでしょうか。0.5 から 5.5 まで 0.5 間隔の実数の 2 乗と 3 乗の値を計算してエディタ mule で書くのでは、計算機の名が泣きます。ここは、やはりプログラムを組んでファイルを作成したいものです。では、プログラム言語は何がいいでしょうか？第 1 候補は前期に学習した **Pascal** でしょう。

Pascal で書いた例：sqx.pas

```
program sqx(output);
   var x: real;

begin
   x := 0.5;
   while(x < 6) do
      begin
         writeln(x:9, x*x:9, x*x*x:9);
         x := x+0.5
      end;
end.
```

これをコンパイルして実行ファイルを作成したいのですが、Linux(GNU) では Pascal のコンパイラがありません。しかし、あらゆるプログラミング言語開発環境が提供されている UNIX では、他の手段があります。**p2c** を使って一端 **C** 言語のソースに変換し、**gcc** でコンパイルするのです。まず、C 言語のソースに変換します。

```
p2c sqx.pas
```

これで **sqx.c** が出来上がっているはずです。これを gcc でコンパイルします。

```
gcc sqx.c -o sqx -lm -lp2c
```

-o *filename* は実行ファイルの名前の指定です。これが無いと **a.out** という実行ファイルが出来てしまいます。**-lm -lp2c** はリンクするライブラリの指定で

す。これは、今のところ必要なオマジナイと考えてください。では、**sqx** を実行してください。データが画面に出力されます。これを、ファイルに保存するには、リダイレクト機能を使って、出力先を変更すれば可能です。

sqx > sqx.dat

どうやら sqx.dat ファイルが出来ました。実は、gpc というコンパイラがありますが、現在インストールされていないので使用出来ません。将来もし gpc が使用可能になったなら *gpc sqx.pas -o sqx* とコンパイルが簡単になります。

しかし、この程度の計算に Pascal や C は大袈裟です。そこで、もっと易しいスクリプト言語を使ってみましょう。例えば、あとの節で詳しく紹介しますが **Perl** でやってみましょう。次のようなスクリプトファイルを sqx.pl として作成してください。

Perl の場合

```
$x = 0.5;
while ($x < 6){
   printf "%.3f %.3f %.3f \n", $x, $x*$x, $x*$x*$x ;
   $x = $x + 0.5;
}
```

このスクリプトを Perl に読み込ませて実行します。

jperl sqx.pl (`> sqx.dat`)

画面に計算結果があらわれましたか？ ( ) 内を追加すると、出力は sqx.dat にリダイレクトされます。コンパイルの手間が無い分、楽です。

寄り道が長すぎました。GNUPLOT に戻りましょう。

5.1.6 範囲 - ∗range [min:max] -

変数の範囲は **set** できます。例えば

set xrange [-1:1]
*set yrange [-pi:2*pi]; set zrange[:0]*

以後はこの変数範囲で図を描きます。が、**plot** 時に範囲指定も可能です。

`gnuplot>` *plot [-2*pi:2*pi][-2:2] sqrt(abs(x))*sin(2*x)*

これだと図がはみ出します。やはり自動設定に戻したいですね。

autoscale の設定

`gnuplot>` *set autoscale y*

これで *y* 軸は図が収まるように自動設定されます。*replot* して確かめてください。

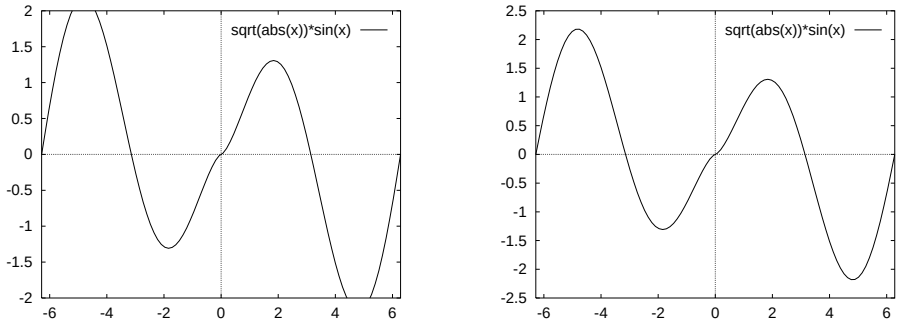

図 32　range 設定の適否：set autoscale で適切な range が設定される

5.1.7　滑らかさ：サンプル数 - samplees -

次のコマンドで描かれた図形を見てください。

```
gnuplot> exp(-0.1*x**2)*sin(5*x)
```

ギザギザです、汚らしいですね。これは描く点の数が不足しているために、折れ線が目だっているのです。もっと点を増やして、滑らかな曲線を描かせましょう。

```
gnuplot> set sample 600
gnuplot> replot
```

だいぶ綺麗になりました。

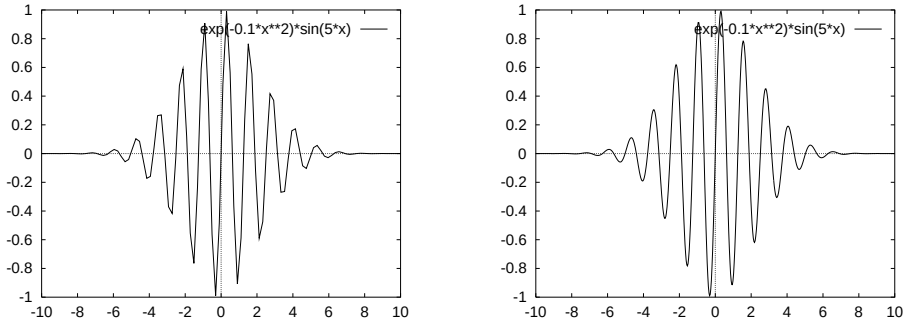

図 33　sample を多くすると曲線が滑らかになる

5.1.8　ヘルプ - help *topics* -

GNUPLOT はオンラインヘルプを備えていますから、これを活用しましょう。疑問が漠然としている場合は

```
gnuplot> help
```

でメニュー形式のヘルプが現れますから、指示に従い一通り読んでみるのもいいかもしれません。また使いこなしていって、内容を知りたい項目名 (topics) がはっきりしている場合には、

gnuplot> *help* トピック名

と直接指定します。

5.2 非対話的な使い方 - スクリプトで制御 -

GNUPLOT のコマンドを並べたファイルを読み込ませて、仕事をさせる事ができます。次の内容で expcos.gnu を作成してください。

```
set isosample 60
set hidden3d
r(x,y)=(x**2+y**2)
set zrange[0:1]
splot [-pi:pi][-pi:pi] exp(-sqrt(r(x,y)))*cos(1.5*r(x,y))
pause -1
```

このスクリプトを GNUPLOT に実行させます。

gnuplot expcos.gnu

すると少し時間がかかるかもしれませんが、下図のように別窓が開かれ, **splot** により関数 $z = f(x, y)$ の俯瞰図が表示されます。set **isosample** n は xy面での格子の細かさを指定します (即ち、$n \times n$ の点で計算することになります)。set **hidden 3d** は陰線処理 (見えない筈の線を隠す処理) を行う意味です。**pause** s は s 秒表示して終了するという意味です。 $s = -1$ では時間が無限になります、この場合 Enter が押されると終了となります。

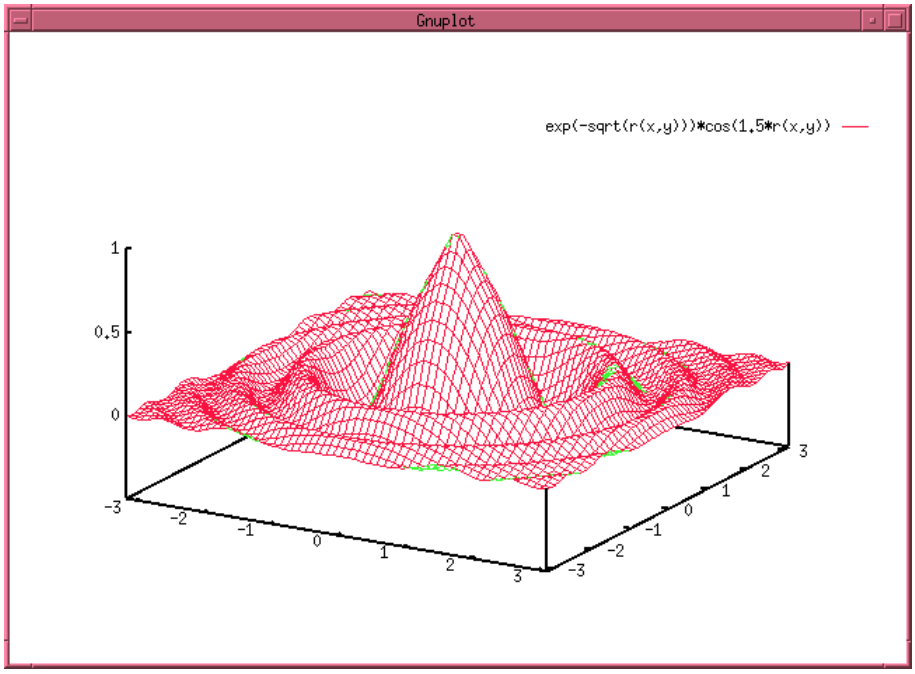

図 34 非対話モードで行った 3 次元プロットの例

5.2.1 tgif の obj 形式で出力する

tgif で読み込んで編集できる形式を指定して出力しましょう。EPS の場合と同様に **term** を tgif に set して、output のファイル名前を指定します。

tgif 形式への出力方法

```
set trem tgif
set output 'filename.obj'
```

練習 自由な場合の固有振動数 ω, 粘性抵抗による減衰率がλ である振動系に振動数 p の外力を加えて強制振動を起こさせる。$\omega^2 - \lambda^2 > 0$ の場合には定常状態における振幅 $C(p)$ は

$$C(p) = \frac{l}{\sqrt{(\omega^2 - p^2)^2 + 4\lambda^2 p^2}}$$

と表される。ここに l は定数である。ω で正規化して改めて、$x = p/\omega$ とおいて、x 依存性部分だけを取り出すと

$$W(x) = \frac{1}{\sqrt{(1 - x^2)^2 + \frac{4\lambda^2}{\omega^2} x^2}}$$

と書かれる。$W(x)$ を$-1 < \log_{10} x < 1$の範囲で、パラメータ $4\lambda^2/\omega^2$ が $1, (0.5)^2, (0.2)^2, (0.1)^2$ の場合について GNUPLOT でグラフを描きなさい。ただし、tgif 上で手を加えて次のような仕上りにしなさい。

出力例：GNUPLOT → Tgif の組み合わせ

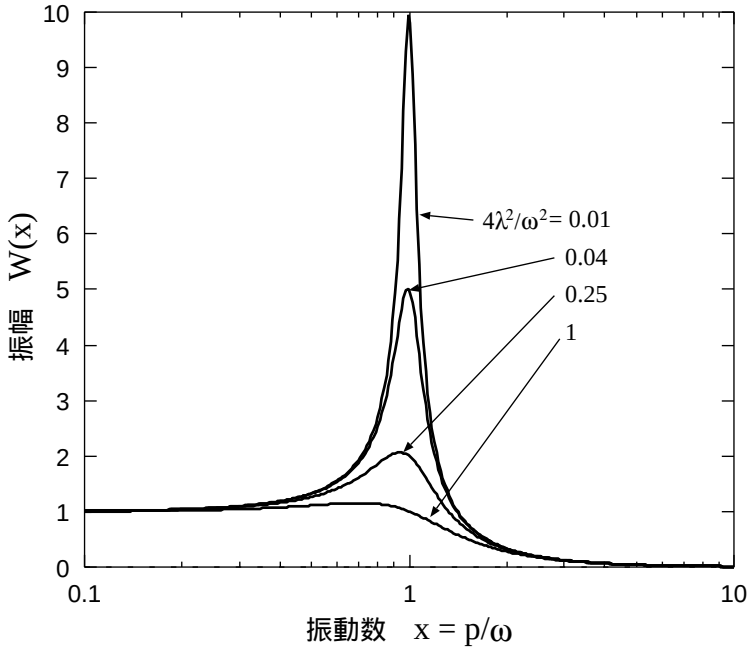

図 35 GNUPLOT → Tgif の組み合わせで作成した例:日本語が可能

5.3 その他に set できる機能

まず set 可能な機能や変数を全て見てみましょう。

```
gnuplot> show all
```

と入力しますと、いやー沢山ありますね。

GNUPLOT の set 可能な機能や変数

```
        G N U P L O T
        Linux 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   Colin Kelley, Thomas Williams

        Send comments and requests for help to info-gnuplot@dartmouth.edu
        Send bugs, suggestions and mods to bug-gnuplot@dartmouth.edu
        autoscaling is  x: ON, y: ON, z: ON
        border is drawn
        boxwidth is auto
        point clip is OFF
        drawing and clipping lines between inrange and outrange points
        not drawing lines between two outrange points
        contour for surfaces are not drawn
        data grid3d is disabled
        mapping for 3-d data is cartesian
        dummy variables are "x" and "y"
        tic format is x-axis: "%g", y-axis: "%g", z-axis: "%g"
        data are plotted with points
        functions are plotted with lines
        grid is OFF
        key is ON
        no logscaling
        offsets are 0, 0, 0, 0
        output is sent to STDOUT
        parametric is OFF
        polar is OFF
        Angles are in radians
        sampling rate is 100, 100
        iso sampling rate is 10, 10
        view is 60 rot_x, 30 rot_z, 1 scale, 1 scale_z
        surface is drawn
        hidden surface is drawn
        size is scaled by 1,1
        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 [-10 : 10]
        yrange is [-10 : 10]
        zrange is [-10 : 10]
        title is "", offset at 0, 0
        xlabel is "", offset at 0, 0
        ylabel is "", offset at 0, 0
        zlabel is "", offset at 0, 0
        zero is 1e-08
        last plot command was:

        Variables:
        pi = 3.14159

        User-Defined Functions:
```

前にも言いましたが項目を help で調べることができます (が英語です)。幾つかについて補足します。

(no)border グラフの周りの枠を描くかどうか。

format 軸の目盛の数値の表記形式を指示します。C 言語に準じていますから、

指定例	意味と例
set format x "%e"	浮動小数点形式 例 1.0E-12, 3.45E+6
set format x "%.3e"	小数点以下 3 桁表示 例 1.002E-12, 3.458E+6
set format xy "%f"	固定小数点形式 例 1.12345, 0.00004
set format xy "%.2f"	小数点以下 2 桁表示 例 1.12, 0.00
set format z "%g"	%f と%e で短く表記可能な方で表示する

(no)grid tic の位置に、格子線を描くかどうか。

(no)key 各プロット毎の名前を (グラフ内右上に) 表示するかどうか。

label 軸に名前を付けます。例 *set xlabel 'Frequency'* 。

(no)logscale 軸を対数にするかどうか。例 *set logscale xy*

angles degrees | radians 極座標系での角度の単位の設定。デフォルトは degree(度) なので、radian に変えたい場合は *set angles radian* と set します。

view 3 次元プロットの場合の視点方向の設定。4 つの値;x 軸まわりの回転角度, z 軸まわりの回転角度, 全体の拡大率, z軸方向のみの拡大率を指定する。例えば *set view 0,0,,* とすると、xy 平面を z 軸から見る (2 次元プロットの視点と同じ) ことになります。また *set view 90,0,,* とすると、xz 平面を y 軸負の側から見ることになります。

size 水平方向と垂直方向の拡大率を設定します。X11 上では水平方向が長いので、正方形にしたい場合には、*set size 0.721,1.0* にしなければなりません。EPS 形式では 10 : 7 がデフォルトのアスペクト比なので、*set size 0.7,1.0* とすると正方形 (アスペクト比が 1:1) になります。

∗tics 目盛の付け方を設定します。第 1 の形式では `set xtics <初期値>,<間隔>,< 終値 >` と記述します。例えば `set xtics 0,0.2,1.0` では `0,0.2,0.4,0.6,0.8,1.0` に目盛がふられます。第 2 の形式では `ytics ("名前"<位置>,"名前"<位置>, ...)` と記述します。例えば `set ytics ("low" 0, "medium" 3.5, "high" 5.0)` や位置のみの `set xtics (1,2,4,7,10,20,40,70,100)` 等です。