

情報リテラシー gnuplot の簡単な使い方

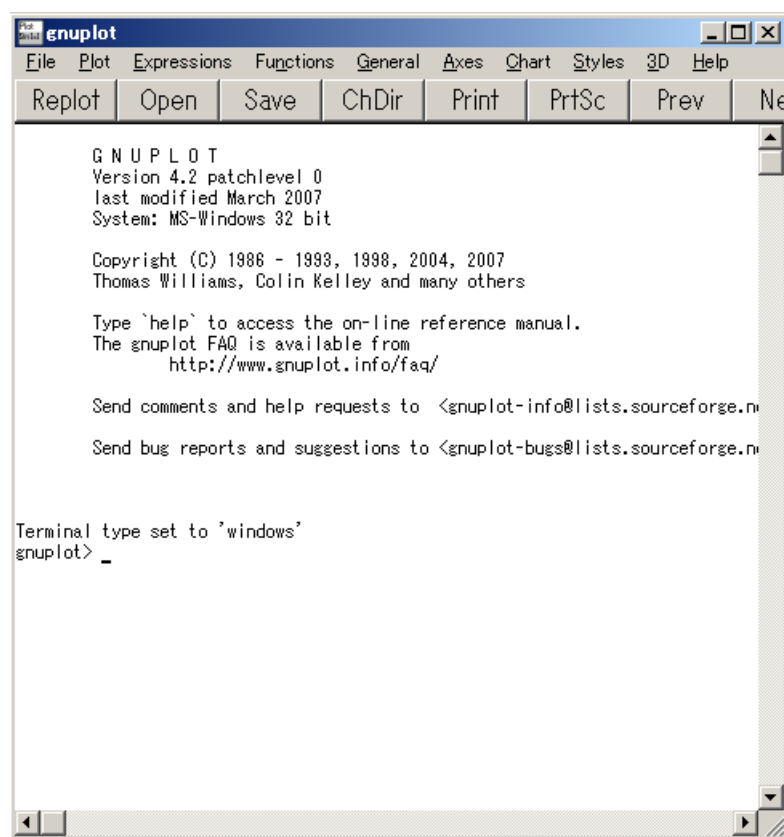
T.A.津崎友一郎

まず始めに

gnuplot を起動してください。

起動すると背景が白で黒い文字の画面が現れると思います。

gnuplot はその画面に命令文を入力することによってグラフを描いたり、計算させることが出来ます。

The image shows a screenshot of the gnuplot application window. The window title is "gnuplot". The menu bar includes "File", "Plot", "Expressions", "Functions", "General", "Axes", "Chart", "Styles", "3D", and "Help". The toolbar contains buttons for "Replot", "Open", "Save", "ChDir", "Print", "PrtSc", "Prev", and "Next". The main text area displays the following information:

```
GNU PLOT
Version 4.2 patchlevel 0
last modified March 2007
System: MS-Windows 32 bit

Copyright (C) 1986 - 1993, 1998, 2004, 2007
Thomas Williams, Colin Kelley and many others

Type `help` to access the on-line reference manual.
The gnuplot FAQ is available from
  http://www.gnuplot.info/faq/

Send comments and help requests to <gnuplot-info@lists.sourceforge.net>
Send bug reports and suggestions to <gnuplot-bugs@lists.sourceforge.net>

Terminal type set to 'windows'
gnuplot> _
```

今回の gnuplot の主な説明はグラフの描き方です。

グラフを描けるようになることで、レポートを提出するときなどきれいなグラフを添えることが出来るようになるようになると思うので、大学4年間、何度も使う機会があるので参考書などを用いているような機能を自分で調べたりなどしてこれからの勉強に役立ててください。

また、gnuplot のように命令文を打ち、その操作を PC にさせるということをこれからする機会が多くなると思うので gnuplot を通じてそんな操作に慣れておくと良いでしょう。

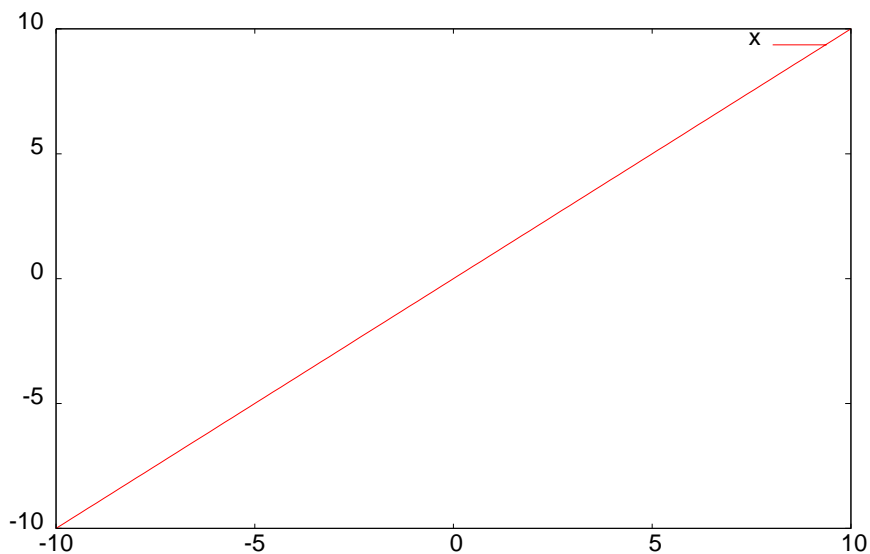
プロットしてみよう

まず

```
gnuplot> plot x
```

と入力してください。

するとこんな画面が出ましたよね。



plot のあとに数式を書くと、その数式がグラフになって表示されます。

また

```
gnuplot> replot
```

あるいは

```
gnuplot> rep
```

などと打つと、前回プロットしたグラフを表示させることができます。

また、

```
gnuplot> replot x+1
```

と入力すると、前回表示させたグラフに続いて、replot の後続く関数を表示させることも出来ます。

複数のグラフを同時に出力してみましょう。

```
gnuplot> plot x-1, x+1
```

と入力してください。

そうすると2つのグラフを表示させることができます。

カンマ (,) で式と式の間を区切ることによって複数のグラフを表せられるようになります。

```
gnuplot> plot x , x+1 , x+2
```

など3つのグラフを同時に出力できます。

足し算は	+
引き算は	-
掛け算は	*
割り算は	/
乗数は	**
平方根は	sqrt(x)
指数関数は	exp(x)
自然対数は	log(x)
サインは	sin(x)
コサインは	cos(x)

です。

変数と関数の定義

```
gnuplot> a=3
```

と入力してください。

そのあとに

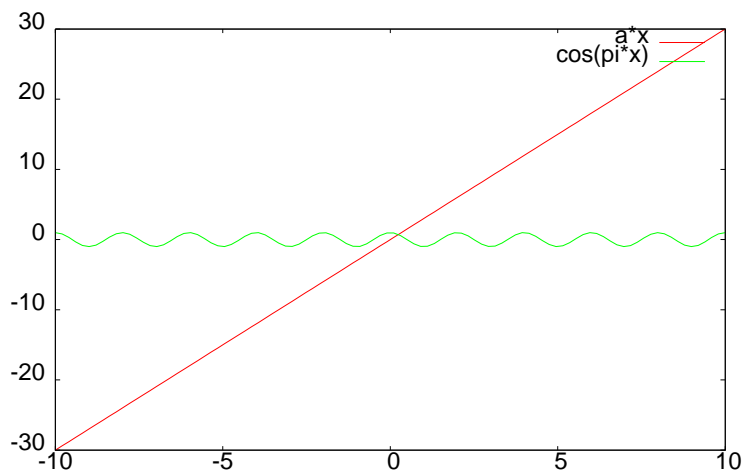
```
gnuplot> plot a*x
```

とすると $3x$ のグラフが出てきます。

こんなふうに定数をきめてからグラフの出力をさせることもできます。

デフォルトとして、円周率 の値が変数 pi に代入されています。

```
gnuplot> plot cos(pi*x)
```



また

```
gnuplot> f(x)=x**2
```

と入力したあとに、

```
gnuplot> plot f(x)
```

と入力すると x^2 のグラフが出てきます。

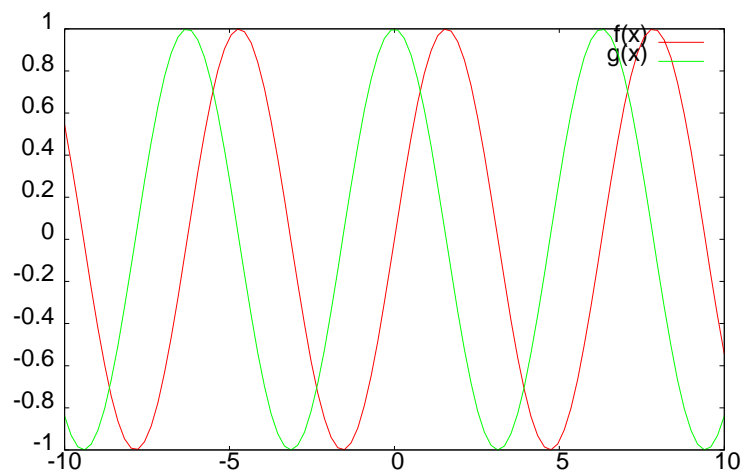
では

```
gnuplot> f(x)=sin(x)
```

```
gnuplot> g(x)=cos(x)
```

```
gnuplot> plot f(x) , g(x)
```

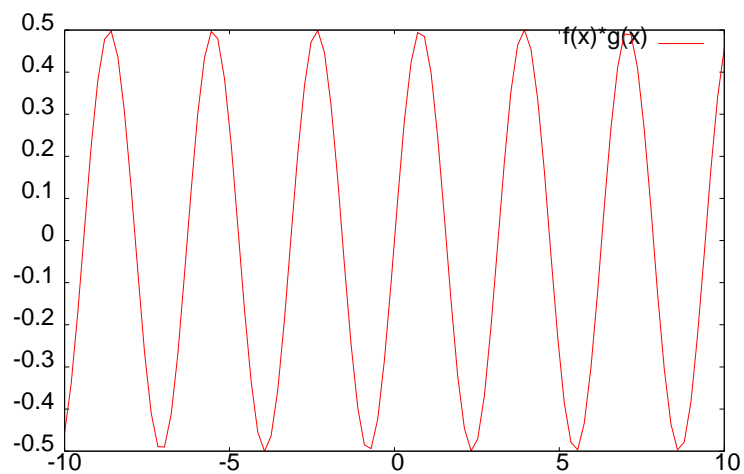
と入力してください。



また、

```
gnuplot> f(x)*g(x)
```

といったグラフも描けます。



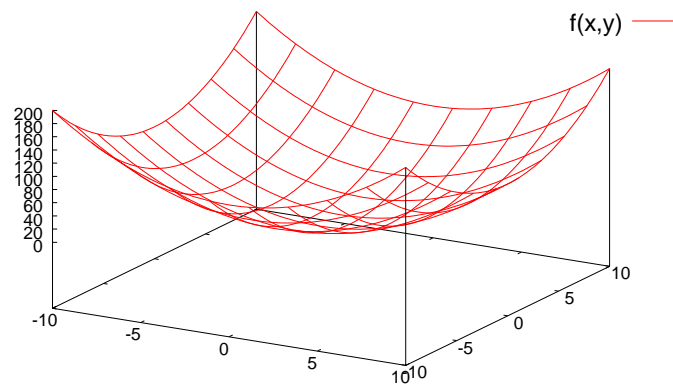
関数のプロット（3次元）

3次元のグラフのプロットも可能です。

その場合は plot ではなく splot になります。

```
gnuplot> f(x,y)=x**2+y**2
```

```
gnuplot> splot f(x,y)
```

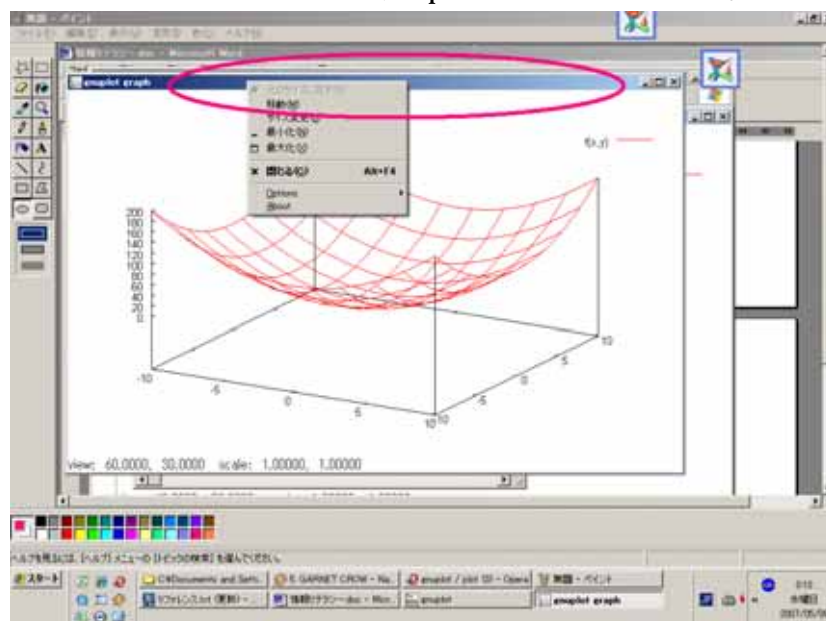


グラフを Microsoft Word に貼り付けてみよう

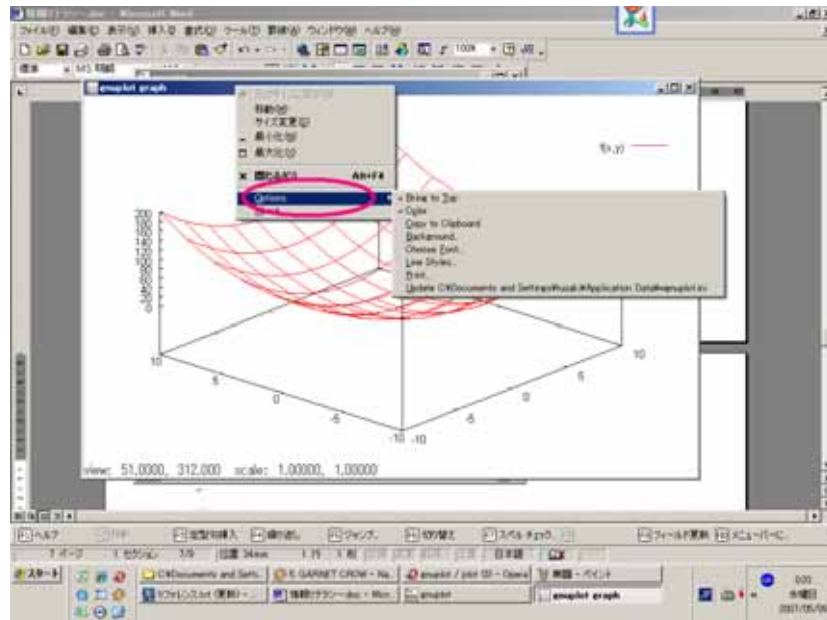
gnuplot で作成したグラフを Word に貼り付けてみましょう。

図の赤丸で囲われた青色のバーを右クリックしてください。

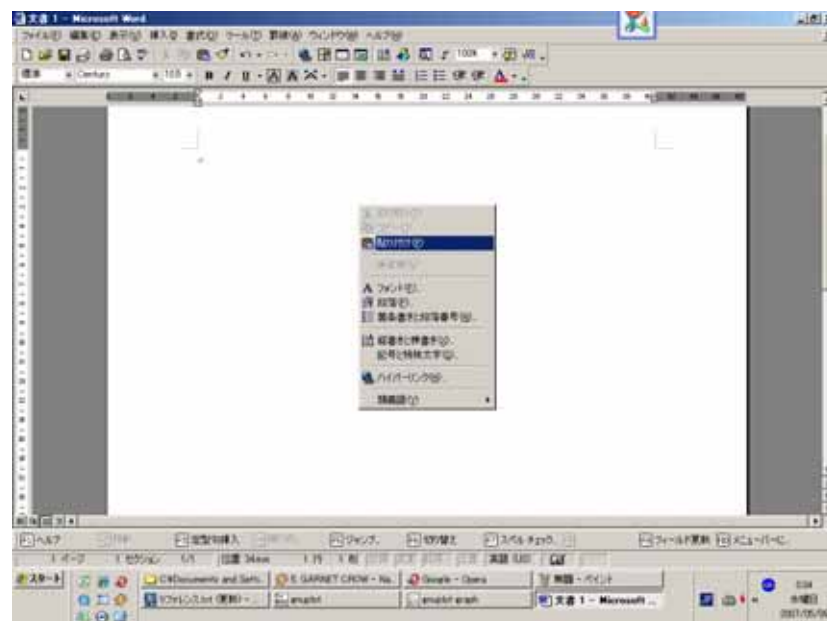
そこをクリックしたらこんな出ますんで、Option を選んでください。



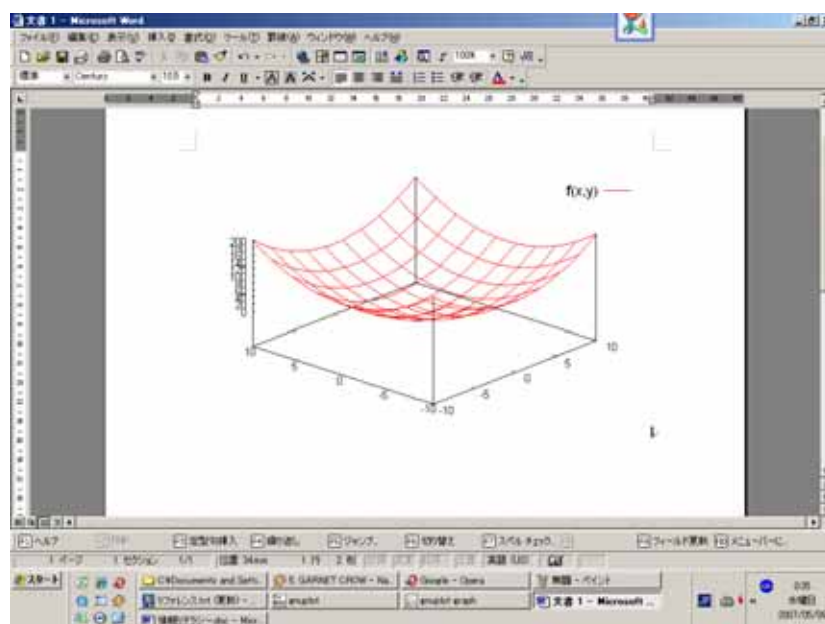
そうすると色々な選択肢がありますが(いろいろ試して遊んでみてください)
Copy to Clipboard を選んでください。



Microsoft Word を起動させてください。
適当な箇所で右クリックし貼り付けを選択してください。



そうすると Word にグラフを貼り付けることができます。



関数電卓としての機能

```
gnuplot> print pi
```

と入力してみましょう。

円周率 の値が出力されます。

gnuplot にはこんな機能もあります。

```
gnuplot> a=142857
```

と入力した後、a の値を確認するために print 文を使って見ましょう。

```
gnuplot> print a
```

としたら確認が出来ます。

```
gnuplot> f(x)=a*x
```

と定義したとき、例えば $x=7$ のときの f の値を確認したいとき

```
gnuplot> print f(7)
```

と入力するとその値が出力されます。

ちなみに、定義された変数の値をすべて確認したいときは

```
gnuplot> show variables
```

と入力すると一覧が見れます。

また、ユーザーによる定義された関数をすべて確認したいときは

```
gnuplot> functions
```


と入力すると確認することができます。

ファイルを読み込んでプロットさせる

gnuplot は数字が並んだファイルを読みこんで、それをプロットさせることができます。とりあえず実際にやってみましょう。

まず、gnuplot の本体があるディレクトリの隣にある demo というディレクトリに移動しましょう。

分からない人は以下の文字をそのまま入力しましょう。

```
gnuplot> cd "../demo/"
```

そこに「world.dat」というファイルがあり、それは数字が並んだファイルです。

それをプロットさせてみましょう。

```
gnuplot> plot "world.dat"
```

world.dat の中にあるファイルの一部を抜粋してみました。

```
140.30 74.99
```

```
137.27 75.44
```

```
138.29 75.98
```

```
146.08 75.29
```

```
147.75 74.73
```

```
145.85 75.06
```

```
141.44 73.88
```

```
141.48 73.84
```

```
0.01 -71.68
```

```
6.57 -70.57
```

```
15.04 -70.44
```

これらは何を表しているかというと、1列目がx軸で、2列目がy軸を表しています。

gnuplot は、その点をグラフに打ち続けていたのです。

```
gnuplot> plot "world.dat" with line
```

あるいは

```
gnuplot> plot "world.dat" w l
```

と入力してください。

そうすると、ファイルの上から順番に点を結びます。

途中でひとつ改行されて何も無い行があると線が途切れます。

他にも、

```
gnuplot> plot "world.dat" with linespoint      (w lp)
```

```
gnuplot> plot "world.dat" with dot           (w d)
```

などがあります。

gnuplot の中には他にも「whale.dat」、「glass.dat」があります。

試して遊んでみてはいかがでしょうか？

これらのファイルの中身には3列、数字が並んでいます。

と、いうことは splot してみると...

より良いグラフを作成するために

作成したグラフに題名をつけたり、縦軸、横軸に名前をつけることが可能です。

他にもいろいろな操作がありますが、まず、よく使う簡単なものだけ紹介します。

それらは命令文 set の後にオプションを書きます。

また、それらのコマンドの解除は set ではなく命令文 unset を用いることによって可能です。

また、その設定を確認するのに show というコマンドもあります。

```
例：gnuplot> set xrange [-1:1]
```

```
例：gnuplot> unset key
```

```
例：gnuplot> show xrange
```

タイトル名をつけるオプション title

グラフにタイトルを表示させます。

ふたつの「」の間にある文字をタイトルとするので、

```
gnuplot> set title "_title_"
```

ではタイトルが_title_と表示されます。

```
gnuplot> unset title
```

と入力するとタイトルを消すことができます。

幅の設定オプション xrange yrange zrange autoscale

```
gnuplot> set xrange [1:10]
```

このような命令文を入力するとグラフに表示される x 軸の幅が 1 から 10 になります。

y 軸、z 軸も同様に以下のようにして決定させることができます。

```
gnuplot> set yrange [-10:10]
```

```
gnuplot> set zrange [0:2*pi]
```

幅の指定を自動に戻したいときにこのコマンドを使います。

```
gnuplot> set autoscale
```

軸に名前をつけるオプション xlabel ylabel zlabel

```
gnuplot> set xlabel "__x__"
```

x 軸の名前をグラフに表示させます。

この場合では__x__と表示されます。

```
gnuplot> set ylabel "__y__"
```

```
gnuplot> set zlabel "__z__"
```

同様に、y 軸では__y__、z 軸では__z__と表示されます。

任意の軸の名前を消すには、例として x 軸の名前を消すには

```
gnuplot> unset xlabel
```

と入力します。

座標軸の表示のオプション zeroaxis xzeroaxis yzeroaxis zzeroaxis

```
gnuplot> set zeroaxis
```

この様に入力すると、プロットさせるすべての座標軸を表示させます。

任意の座標軸にだけ座標軸の表示を命令させるのは以下の通りです。

```
gnuplot> set xzeroaxis          x 座標軸を表示させる。
```

```
gnuplot> set yzeroaxis          y 座標軸を表示させる。
```

```
gnuplot> set zzeroaxis          z 座標軸を表示させる。
```

表示を消すには unset を用いればよい。

```
gnuplot> unset zeroaxis
```

目盛を表示させるオプション grid

```
gnuplot> set grid
```

このように入力するとグラフが目盛付きで表示されます。

表示を消すには

```
gnuplot> unset grid
```

と、入力すればよいです。

軸の目盛の幅を変更させるオプション xtics yticsztics

```
gnuplot> set xtics 0.1
```

と入力すると 0.1 刻みで目盛が表示されます。

y 軸 z 軸についても同様の操作で出来ます。

凡例の表示についてのオプション key

デフォルトの状態ではグラフを表示させるとき、右上にある関数と線の色の照らし合わせのことが凡例です。

凡例をグラフの外側に表示させることができます。

```
gnuplot> set key inside
```

凡例を消すには

```
gnuplot> unset key
```

とすれば可能です。また表示させる場所を操作することが可能です。

例：	<pre>gnuplot> set key default</pre>	適当な位置に表示させる。
	<pre>gnuplot> set key left</pre>	左側に。
	<pre>gnuplot> set key right</pre>	右側に。
	<pre>gnuplot> set key top</pre>	上側に。
	<pre>gnuplot> set key bottom</pre>	下側に。
	<pre>gnuplot> set key outside</pre>	図の外側に。
	<pre>gnuplot> set key inside</pre>	図の内側に。
	<pre>gnuplot> set key below</pre>	図の下側の外に。
	<pre>gnuplot> set key 1,1</pre>	座標(1.1)の場所に表示させる。

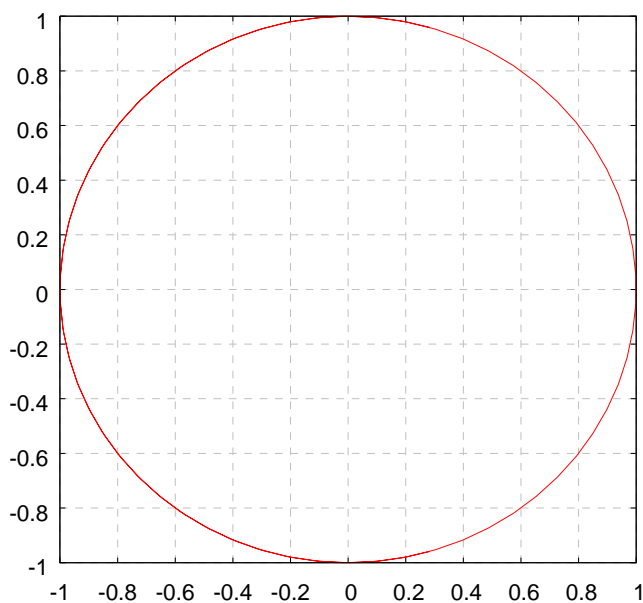
幾つかのものは組み合わせて指定できます。

```
gnuplot> set key outside left bottom
```

 なら、左外側の下の方に凡例が表示されま
す。

媒介変数の使い方

この様なグラフを作成したいときはどうしたらよいでしょうか？



これは原点を中心とした半径 1 の円です。

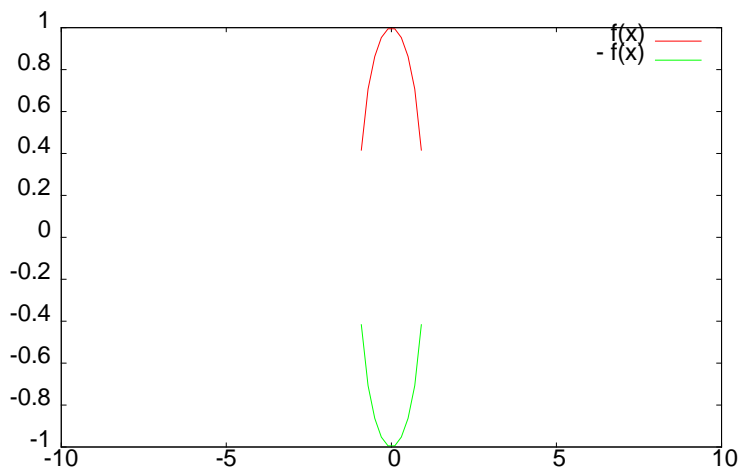
方程式 $x^2 + y^2 = 1$ より、

$f(x) = \sqrt{1-x^2}$ として、

```
gnuplot> f(x)=sqrt(1-x**2)
```

```
gnuplot> plot f(x) , -f(x)
```

この様にプロットした場合、下の図のようにきれいなグラフを描くことが出来ません。



そんなときは媒介変数を使います。

$x = \cos(t)$

$y = \sin(t)$

と置き、gnuplot に媒介変数表示をさせることが出来たらきれいな円グラフを描けるのではないのでしょうか？

```
gnuplot> set parametric
```

あるいは

```
gnuplot> set para
```

と入力すると、gnuplot は媒介変数モードになります。

媒介変数モードになると、plot コマンドを用いる 2 次元座標では媒介変数 t を用います。

```
gnuplot> f(t)=cos(t)
```

```
gnuplot> g(t)=sin(t)
```

```
gnuplot> plot f(t) , g(t)
```

円が描けましたか？

次は 3 次元での媒介変数にチャレンジしてみましょう。

原点を中心とする半径 1 の球体を描いてみます。

$$x = \cos \varphi \sin \theta$$

$$y = \sin \varphi \sin \theta$$

$$z = \cos \theta$$

これを媒介変数表示させてみましょう。

3次元での媒介変数は u と v です。

```
gnuplot> x_para(u,v) = f(u)*g(v)
```

```
gnuplot> y_para(u,v) = g(u)*g(v)
```

```
gnuplot> z_para(u) = f(u)
```

これまで入力した関数に間違いがないか確認してみましょう。

```
gnuplot> show functions
```

ではプロットしてみましょう。

```
gnuplot> splot x_para(u,v) , y_para(u,v) , z_para(u)
```

これで、球体が描けると思います。

$x_para(u,v), y_para(u,v), z_para(u,v)$ —

