

GNU PLOT の簡単な使い方

GNU PLOT は、コマンド入力方式の対話的なグラフ描画プログラムです。

1 最小限操作

1.1 起動と終了

シェル端末上で (以後の説明では「%」をシェルのプロンプトとします) `gnuplot` とタイプすれば起動します。 (注意、コマンド対話的なプログラムですから `&` をつけないで起動して下さい)

—— 起動 ——

```
% gnuplot
```

起動に成功すると GNU PLOT のプロンプト「`gnuplot>`」が表示されて GNU PLOT のコマンドが受け付けられる状態になります。 終了コマンドは「`exit`」です。 GNU PLOT のプロンプト「`gnuplot>`」に対し、`exit` とタイプすれば終了します。

—— 終了 ——

```
gnuplot> exit
```

1.2 データファイルを使ったグラフの描画

例えば

—— sample1.dat ——

```
-2 1
-1 2
0 3
1 4
2 2
```

の様なデータファイルを用意しておいて、「`plot 'ファイル名' with lines`」とタイプすれば描画ウィンドウに図 1 の様なグラフが表示されます。

—— グラフ描画 ——

```
gnuplot> plot 'sample1.dat' with lines
```

1.3 グラフの保存

保存したいグラフを描画した後に、「`set terminal postscript`」「`set output '保存ファイル名'`」「`replot`」とタイプすれば「保存ファイル名」と言うファイルに `postscript` 形式の画像ファイルとして保存されます。 (注意、「`set terminal postscript`」をタイプした時点で描画ウィンドウが閉じます。 復帰させる方法を覚えるまではそのまま GNU PLOT を終了する様にしてください。)

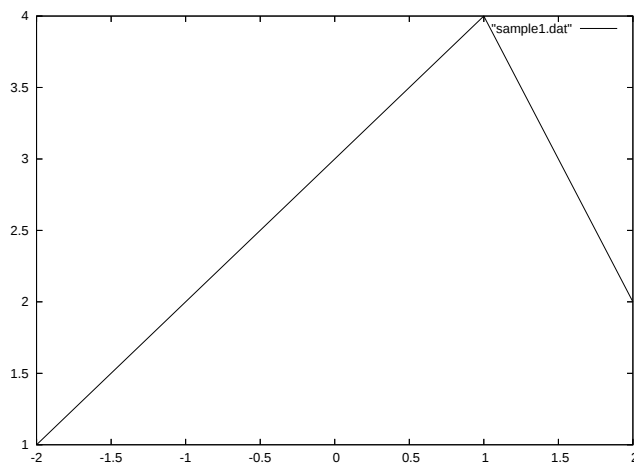


図 1: plot 'sample1.dat' with lines

—— グラフの保存 ——

```
gnuplot> set terminal postscript
Terminal type set to 'postscript'
Options are 'landscape noenhanced monochrome blacktext \
    dashed dashlength 1.0 linewidth 1.0 defaultplex \
    palfuncparam 2000,0.003 \
    butt "Helvetica" 14'
gnuplot> set output 'hoge.ps'
gnuplot> replot
gnuplot> exit
```

postscript 形式の画像ファイルは Ghostview 等の画像ソフトで確認したり印刷したり出来ます。Ghostview は「% gv」で起動します。(Ghostview の詳しい使い方は「% man gv」で調べられます。)

—— 確認 ——

```
% gv hoge.ps &
```

2 データファイルの形式と plot のオプション

plot で用いるデータファイルは、空白と改行で区切られた数値列にしなければなりません。(「#」で始まる行はコメントとして無視されます。)

—— sample2.dat ——

```
# #で始まる行はコメント
-2 1 1
-1 2 1
0 3 0
1 4 0
2 2 1
```

sample2.dat の様に縦に 3 列以上のデータを持つファイルで「plot 'sample2.dat' with lines」とした場合 3 列目以降を無視し、1 列目を x 座標、2 列目を y 座標としたグラフを描画します。(図 1 と同じグラフになります) plot コマンドに using 1:3 というオプションをつけた場合、今度は 1, 3 列目以外を無視し、1 列目を x 座標、3 列目を y 座標としたグラフ (図 3) を描画します。

1 列目-3 列目グラフ描画

```
gnuplot> plot 'sample2.dat' using 1:3 with lines
```

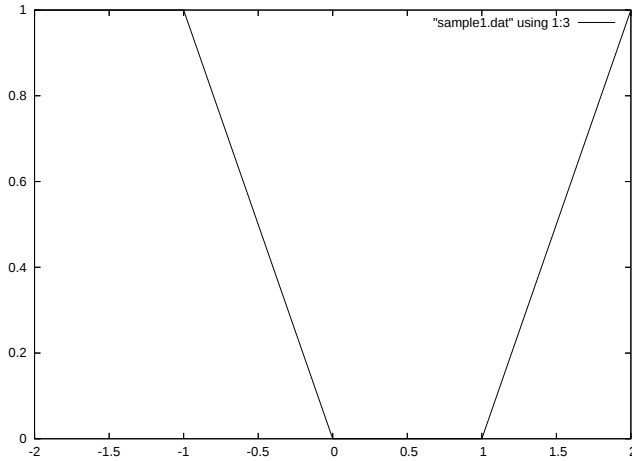


図 2: plot 'sample2.dat' using 1:3 with lines

描画コマンドの「with lines」の部分は”折れ線グラフ”を意味し, 「with points」で標本点のみ「with linespoints」標本点つき折れ線「with impulses」棒グラフ等が指定できます. 各々「w l」「w p」「w lp」「w i」という省略形でも指定できます. 完全に省略した場合「with points」を指定した事になります

「,」区切りで複数のデータを指定してやる事により複数のグラフを描画できます.

複数のグラフの描画

```
gnuplot> plot 'sample2.dat' using 1:3 w l, 'sample2.dat' using 1:2 with linespoints
```

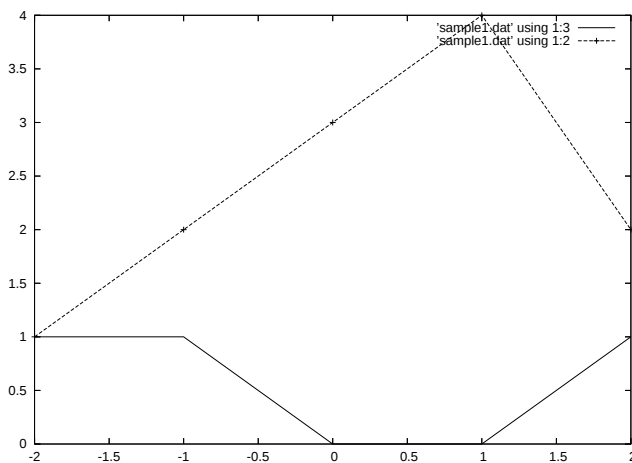


図 3: plot 'sample2.dat' using 1:3 with lines

3 おまけ

この授業では、数値解析プログラムの出力の視覚化の為に GNUPLOT を使用するので、通常、データファイルは (演習で作成した) プログラムによって生成します。

3.1 リダイレクト復習

プログラムによってデータファイルを作成するには、プログラム側では標準出力 (端末画面への表示, `printf` 文等) へ書き出し、実行時にシェルのリダイレクト機能を用いてファイルに保存するのが簡単です。何らかの標準出力 (端末画面) への出力をする実行ファイルに対し「実行ファイル名 > 保存ファイル名」と言う形で実行してやる事により、通常端末画面に出力される結果を指定したファイルに保存できます。

———— リダイレクト ————

```
% ./a.out > 'data.dat'
```

3.2 GNUPLOT の非対話的使用

C で書いたプログラム等で自動生成されるデータファイルを使って描画をする場合、GNUPLOT に対し同じ操作を何度もすることになります。

———— gnuplotcmd ————

```
set terminal postscript
set output 'graph.ps'
plot 'data.dat' with lines
```

そのような場合には `gnuplotcmd` の様にファイルにコマンドを記述しておき、GNUPLOT の起動時にそのファイルを指定する事により、コマンド入力の手間を省く事ができます。

———— 非対話的使用 ————

```
% gnuplot gnuplotcmd
```

4 参考

本格的な使用法はマニュアルを参考にしてください。

公式 <http://www.gnuplot.info/>

日本語化されたリファレンスマニュアル <http://takeno.iee.niit.ac.jp/%7Ef00/gp-jman/gp-jman.html#manjp>