

数値計算法

第五回

グラフィック・ユーティリティ —*gnuplot*—

若狭 智嗣

粒子物理学講座

レポート課題について

- ・ 51ページの練習を第一回レポート問題とする。

練習

本章で解説したプログラムを改良し、平均値と標準偏差を出力するプログラムを作成せよ。また、それぞれのデータの偏差値 (平均が 50、標準偏差が 10 になるように変換した値) を出力するプログラムを作成せよ。

平均値 $\bar{x} = \frac{1}{N} \sum_{i=1}^N x(i)$

分散 $\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x(i) - \bar{x})^2$

標準偏差 $\sigma = \sqrt{\sigma^2}$

偏差値

・ 偏差値 y は、データ x を以下の条件を満たすように変換した値

- 平均値: $\bar{y} = 50$ 標準偏差: $\sqrt{\sigma_y^2} = 10$

・ 線形変換 $y = ax + b$ を考える

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y(i) = \frac{1}{N} \sum_{i=1}^N \{ax(i) + b\}$$

$$= a \frac{1}{N} \sum_{i=1}^N x(i) + b = a\bar{x} + b = 50$$

$$\sigma_y^2 = \frac{1}{N} \sum_{i=1}^N (y(i) - \bar{y})^2 = \frac{1}{N} \sum_{i=1}^N (ax(i) - a\bar{x})^2$$

$$= a^2 \frac{1}{N} \sum_{i=1}^N (x(i) - \bar{x})^2 = a^2 \sigma_x^2 = 100$$

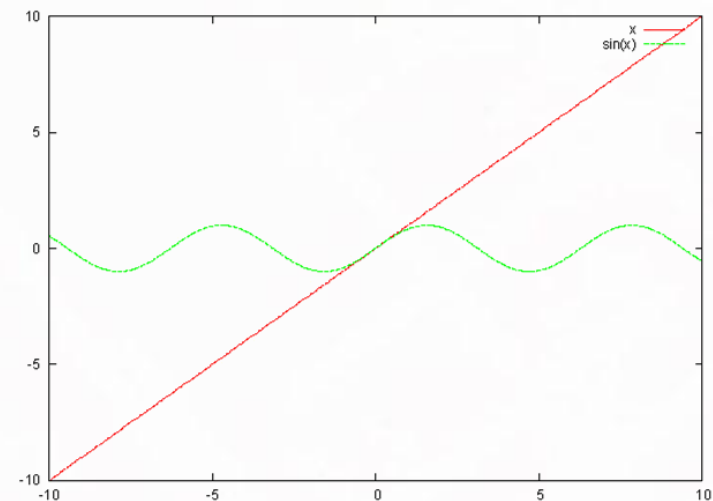
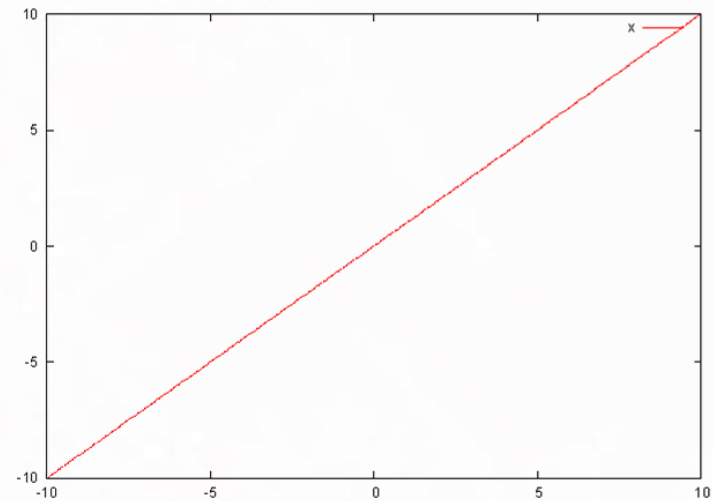
$$a = \frac{10}{\sigma_x}, \quad b = 50 - \frac{10\bar{x}}{\sigma_x} \quad \longrightarrow \quad y = \frac{10(x - \bar{x})}{\sigma_x} + 50$$

Gnuplotの使い方

- ・ gnuplotとは
 - 数式をグラフにする道具
 - データ(表)をグラフにする
- ・ 数値計算や実験データの表示に使える
 - 学生実験のレポート作成や、数値計算結果の表示に役立つ
- ・ gnuplotの起動と終了
 - 起動方法
 - ・ % gnuplot
 - 最後に & を付けない
 - ktermはgnuplotの端末(コマンドを入力)になる
 - 終了方法
 - ・ gnuplot> exit または quit

式のグラフ

- ・ 書式(単一のグラフ)
 - `gnuplot> plot 式` (変数はxで表す)
 - 例
 - ・ `gnuplot> plot x`
 - ・ `gnuplot> plot sin(x)`
(以前描いたグラフはクリアされる)
- ・ 書式(複数のグラフ)
 - `gnuplot> plot 式1, 式2, ...`
(式をカンマで区切る)
 - 例
 - ・ `gnuplot> plot x, sin(x)`
- ・ Xの範囲は[-10,10]、yの範囲はグラフがうまく収まるように自動調整(デフォルト)



演習 1

・ 次のグラフを描け

— $x^4 - 75x^2$

多項式

— $\cos x$

三角関数

— e^x

指数関数

— $\log x$

対数関数

— $\exp(-x^2)$

ガウス関数1

— $\exp(-0.1x^2)$

ガウス関数2

・ 上記の2つのガウス関数を同時に描け(2本の線を同時に描け)

データの表示(2次元)

データの書式

- 1行にxの値とyの値がスペースで区切られたものを用意
- 1文字目が#で始まる行はコメント行(無視される)
- 例
 - ・ `/home/teacher/z6wt01 in/SAMPLE/test1. dat`

```
# test1.dat
# x      y
-10.0 -0.839
-9.0  -0.911
-8.0  -0.146
-7.0   0.754
```

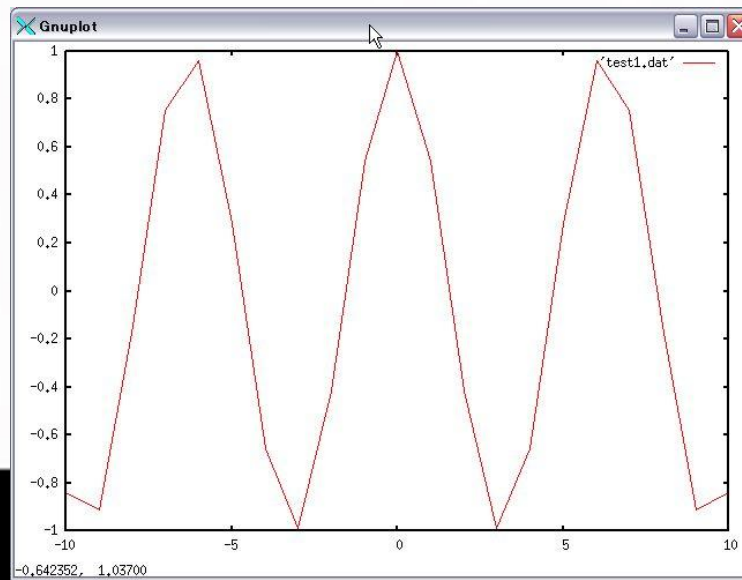
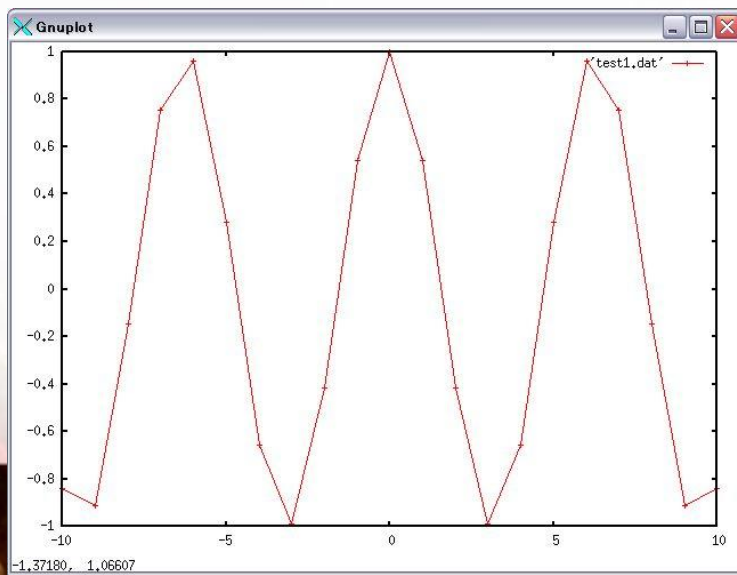
書式

- `gnuplot> plot 'ファイル名' (または"ファイル名")`
- 例
 - ・ `gnuplot> plot 'test1.dat'`
(test1.datがgnuplotを起動したディレクトリにある場合)
 - ・ `gnuplot> plot '/home/06nen/sc010000/test1.dat'`
(test1.datがgnuplotを起動したディレクトリにない場合)

データを線で繋ぐ

書式

- `gnuplot> plot 'ファイル名' with 線種`
- 線種
 - `linespoints` データ点とデータを結んだ線を表示
 - `lines` データを結んだ線のみを表示
- 例
 - `gnuplot> plot 'test1.dat' with linespoints`
 - `gnuplot> plot 'test1.dat' with lines`



3次元のデータプロット

- 書式(式の場合)

- `gnuplot> splot 式`
(変数はx,yで表す)

- 例

- ・ `gnuplot> splot x**2+y**2`

- 書式(データの場合)

- `gnuplot> splot 'ファイル名' with 線種`

- データファイルの書式

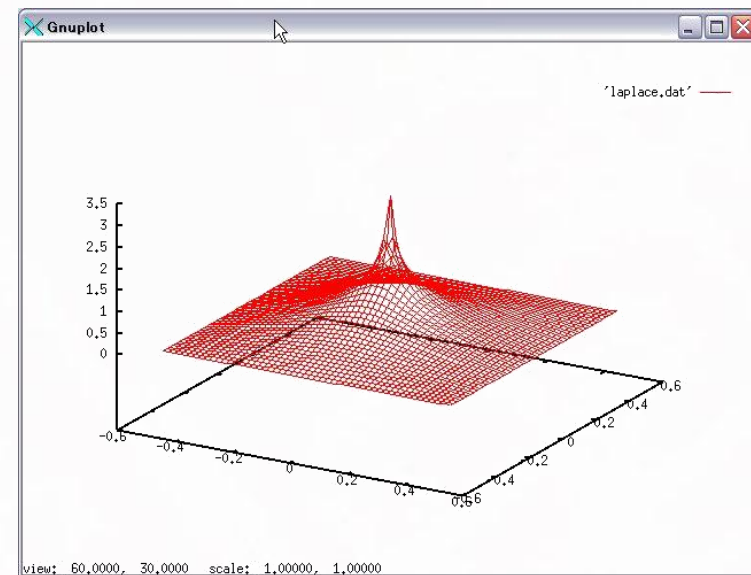
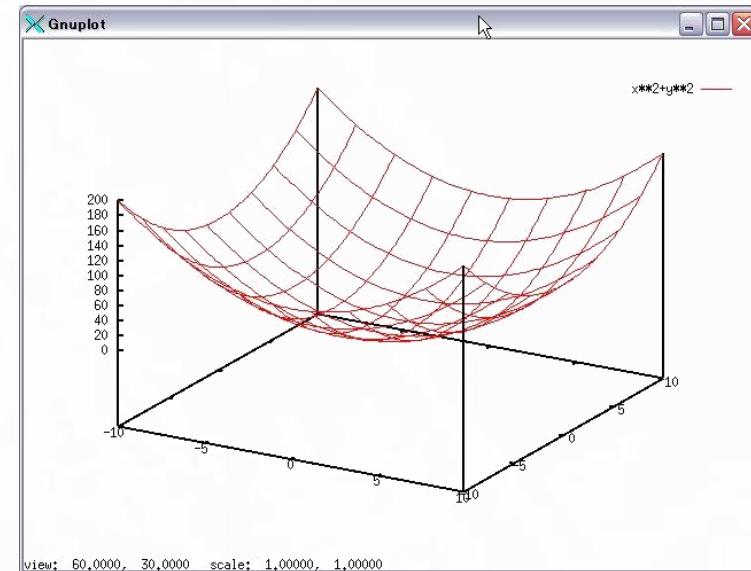
- ・ 講義ノート参照

- 例

- ・ `gnuplot> splot 'laplace.dat' with line`

- laplace.datは以下にある

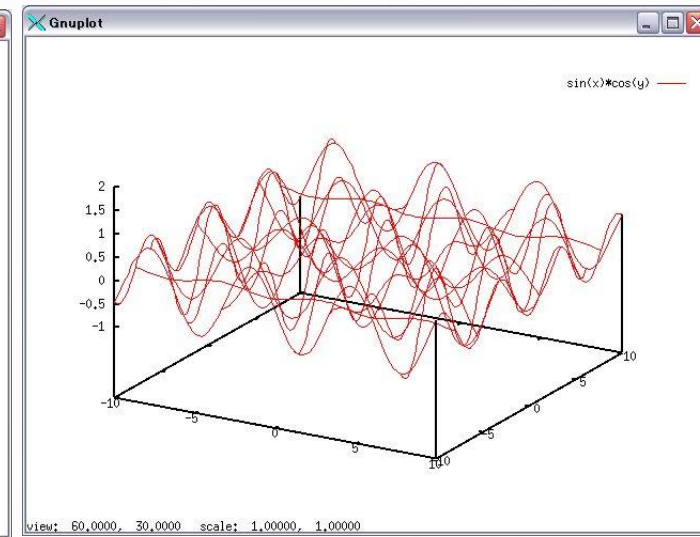
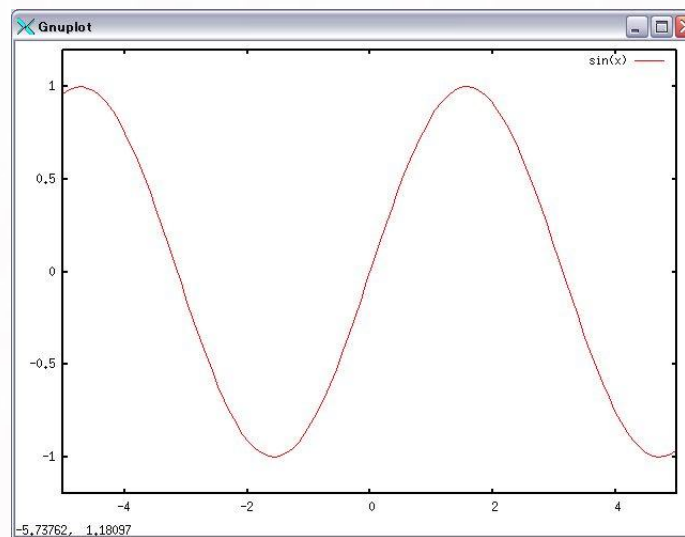
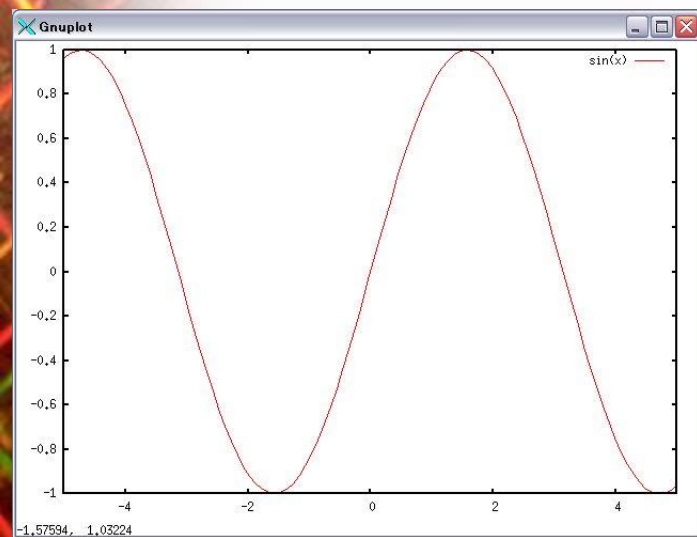
- ・ `/home/teacher/z6wt01 in/SAMPLE/laplace.dat`



グラフのカスタマイズ

グラフの上下限の指定の書式

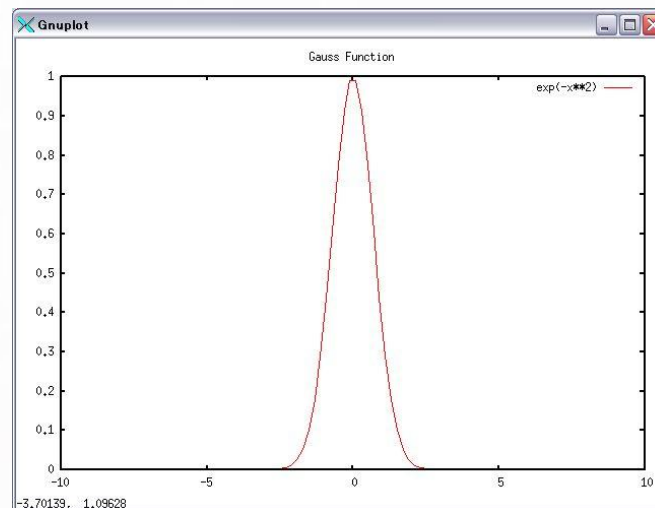
- `gnuplot> plot [xの下限:xの上限] [yの下限:yの上限] 式`
- `gnuplot> splot [xの下限:xの上限] [yの下限:yの上限] [zの下限:yの上限] 式`
 - ・ データを表示する場合も同様
- 例
 - ・ `gnuplot> plot [-5:5] sin(x)`
 - ・ `gnuplot> plot [-5:5] [-1.2:1.2] sin(x)`
 - ・ `gnuplot> splot [] [] [:2] sin(x)*cos(y)`



タイトルや軸にラベルを付ける

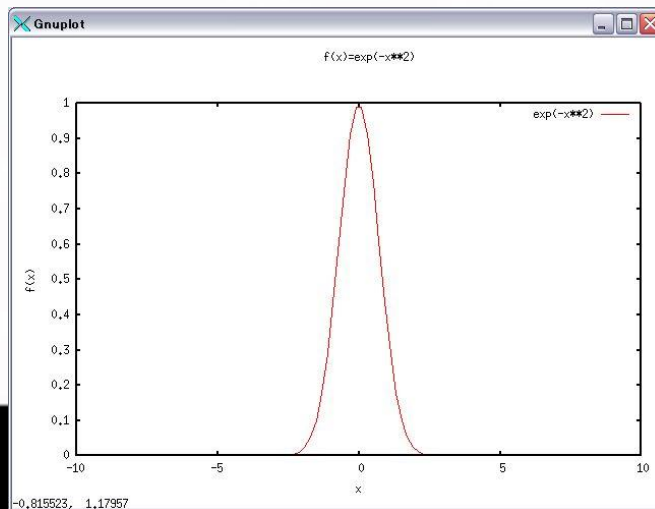
タイトルの付け方

- `gnuplot> set title 'タイトル' 横へのオフセット,縦へのオフセット`
- ・ オフセットは中央上部からのオフセット文字数を指定
- 例
 - `gnuplot> set title 'Gauss Function'`
 - `gnuplot> plot exp(-x**2)`



ラベルの付け方

- `gnuplot> set xlabel 'x軸のラベル' 横へのオフセット,縦へのオフセット`
- `gnuplot> set ylabel 'y軸のラベル' 横へのオフセット,縦へのオフセット`
- 例
 - `gnuplot> set title 'f(x)=exp(-x**2)'`
 - `gnuplot> set xlabel 'x'`
 - `gnuplot> set ylabel 'f(x)'`
 - `gnuplot> plot exp(-x**2)`



演習2

- ・ 次のグラフを描け

- － $\exp\left(-\frac{x}{2}\right)\sin(3x)$

減衰振動の振幅

- － $\exp\left(-\frac{x}{2}\right)$

減衰振動の振幅の上限

- － $-\exp\left(-\frac{x}{2}\right)$

減衰振動の振幅の下限

- － $2\exp(-x) - \exp(-2x)$

過減衰の振幅

- ・ 上記のグラフを同時に描け(4本の線を同時に描け)

- ・ 横軸の領域を $[0,10]$ にして、上記のグラフを同時に描け

- ・ さらに、以下のことを行え

- － 横軸に “Time (s)” というラベルを付けよ

- － 縦軸に “Amplitude” というラベルを付けよ

曲線のラベルの変更

gnuplotでは、式やデータファイル名がグラフ右上に表示される

- キーとよぶ

- 表示されるキー(名)は変更可能で

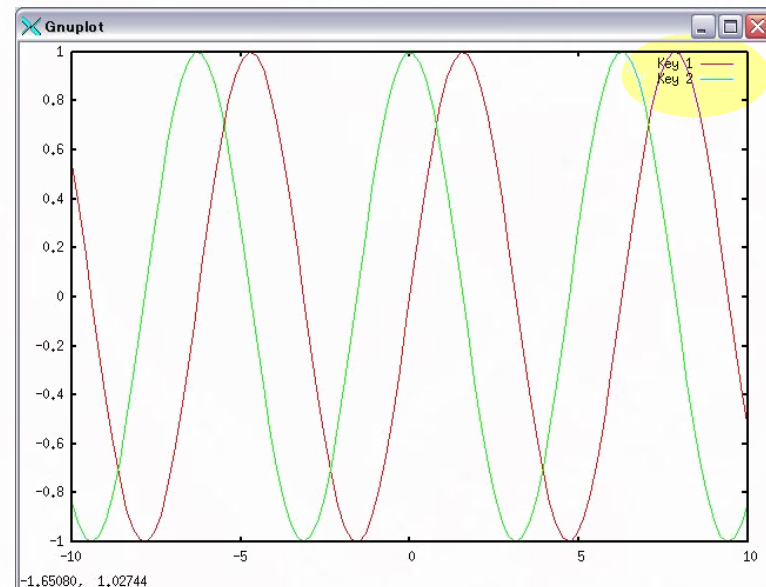
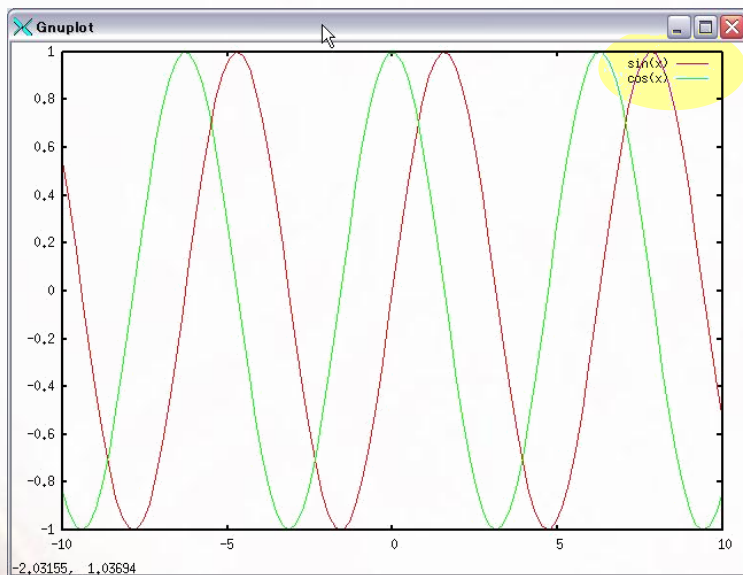
・ `gnuplot> plot 式 title 'キー'`

・ `gnuplot> splot 式 title 'キー'`

- 例

・ `gnuplot> plot sin(x),cos(x)`

・ `gnuplot> plot sin(x) title 'Key 1',cos(x) title 'Key2'`



対数グラフ

- 各軸を対数スケールにする書式

- `gnuplot> set logscale 軸 底`

- ・ 軸: `x y z xy yz xz xyz` のいずれか

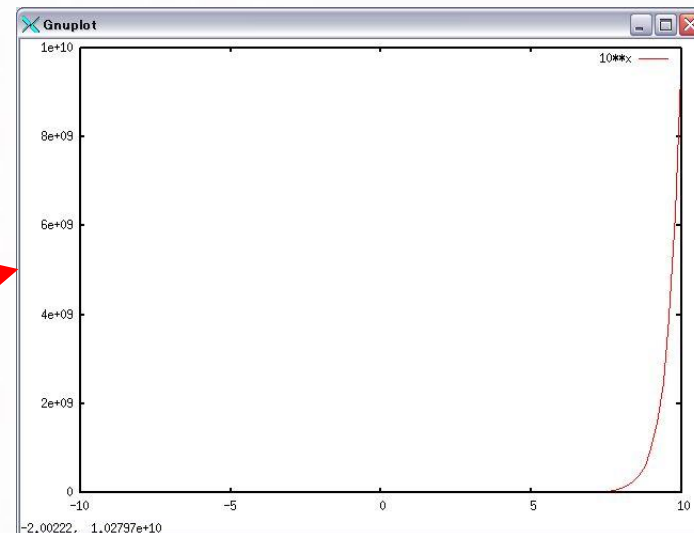
- ・ 底: 省略すると10

- 例

- ・ `gnuplot> plot 10**x`

- ・ `gnuplot> set logscale y`

- ・ `gnuplot> plot 10**x`

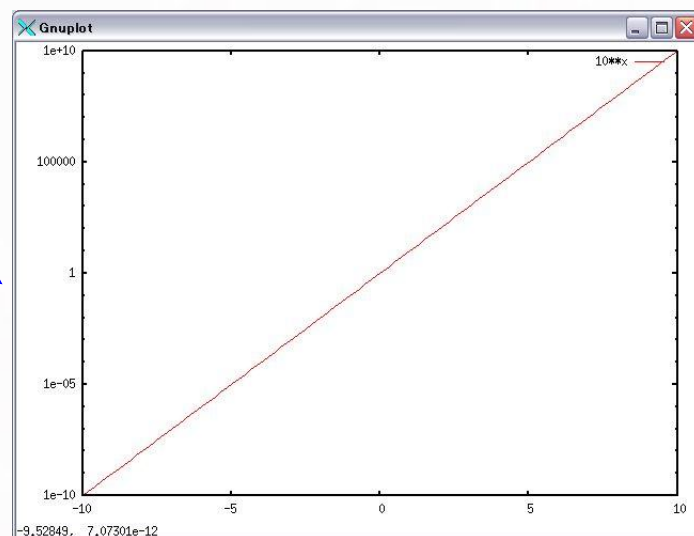


- 各軸を通常のスケールに戻す書式

- `gnuplot> unset logscale 軸`

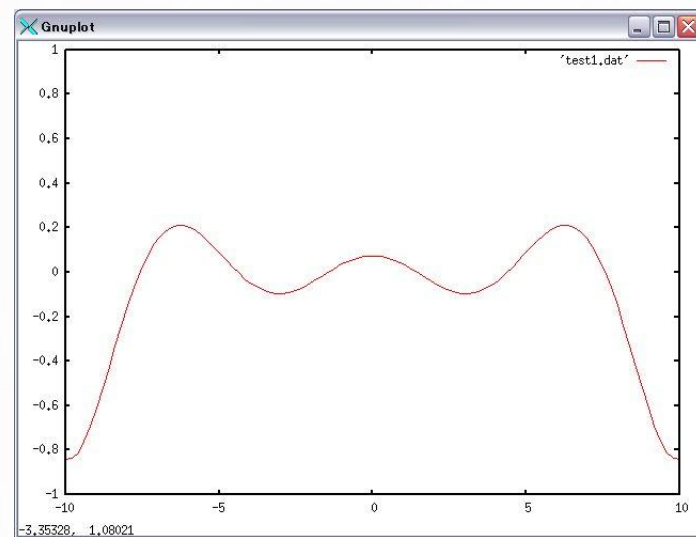
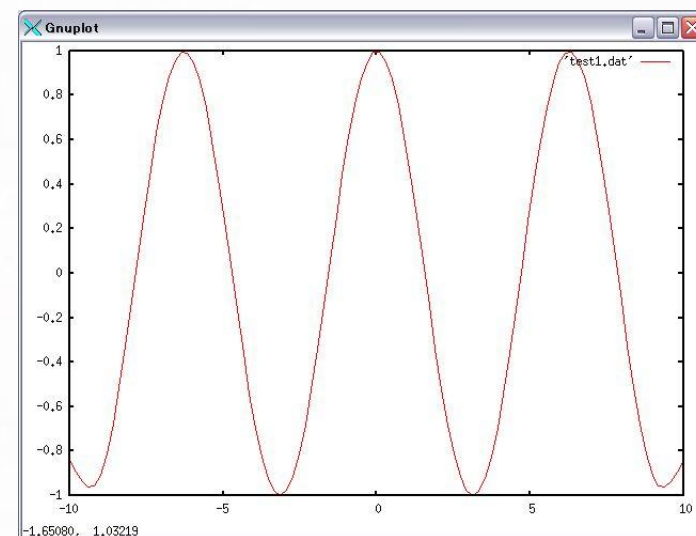
- ・ 軸: `x y z xy yz xz xyz` のいずれか

- ・ 軸を省略すると、全ての軸が通常のスケールになる



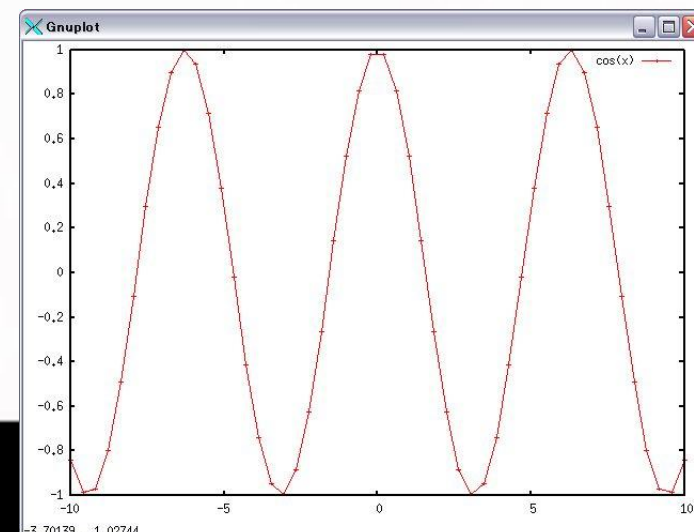
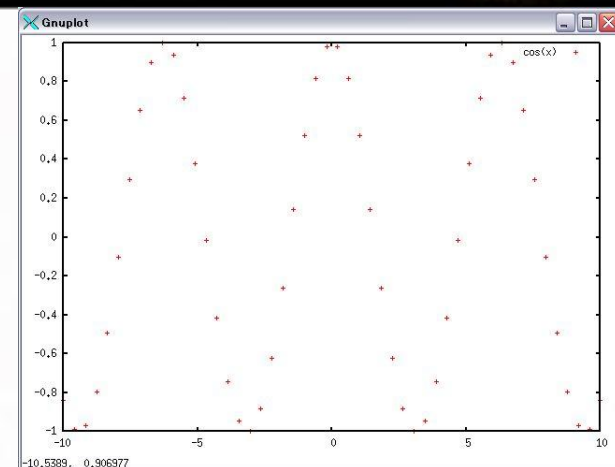
データを補間する

- データ点と点の間を滑らかな曲線で結ぶ書式
 - gnuplot> plot 'ファイル名' smooth 補間方法
 - gnuplot> splot 'ファイル名' smooth 補間方法
 - 補間方法(詳細は各人調べること)
 - csplines スプライン補間
 - bezier ベジエ曲線
 - 例
 - gnuplot> plot 'test1.dat' smooth csplines
 - gnuplot> plot 'test1.dat' smooth bezier
 - test1.datは以下にある
`/home/teacher/z6wt01 in/SAMPLE/test1. dat`



グラフの表示形式

- 関数やデータをラインのみで表示する場合
 - `gnuplot> plot 式(または'ファイル名') with lines`
 - `gnuplot> splot 式(または'ファイル名') with lines`
- 関数やデータを点(マーク)のみで表示する場合
 - `gnuplot> plot 式(または'ファイル名') with points`
 - `gnuplot> splot 式(または'ファイル名') with points`
 - 式の場合は、標準で等間隔にサンプル点が100点選ばれる
 - ・ 変更するには→`gnuplot> set samples サンプル点の数`
- 関数やデータをラインと点で表示する場合
 - `gnuplot> plot 式(または'ファイル名') with linespoints`
 - `gnuplot> splot 式(または'ファイル名') with linespoints`
 - 例
 - ・ `gnuplot> set samples 50`
 - ・ `gnuplot> plot cos(x) with linespoints`
(点と点の間は直線で結ばれる)



ヒストグラム表示

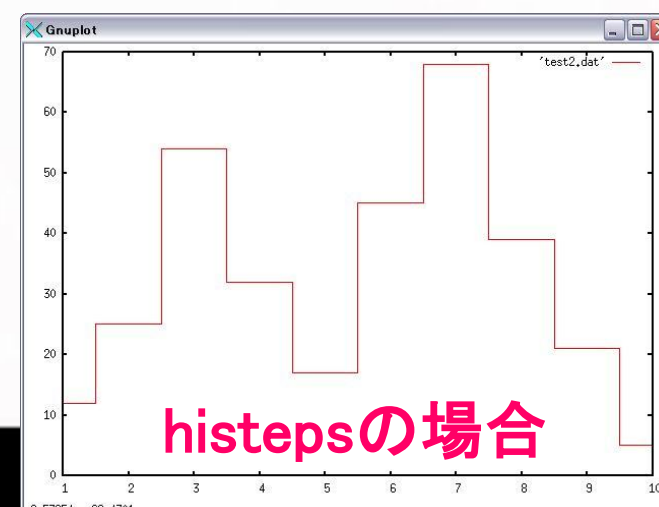
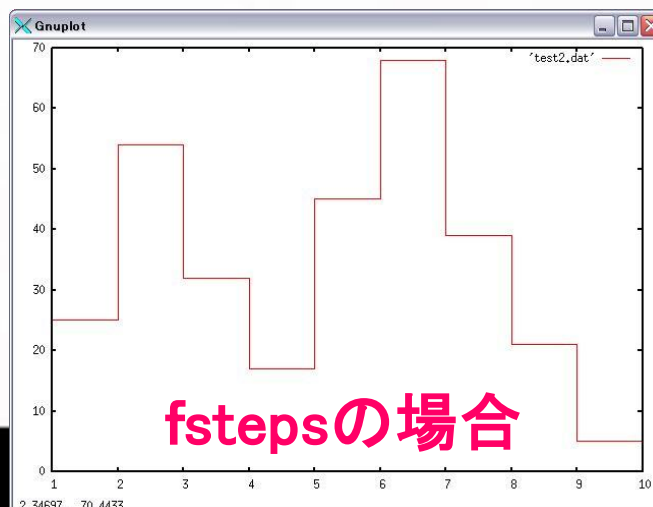
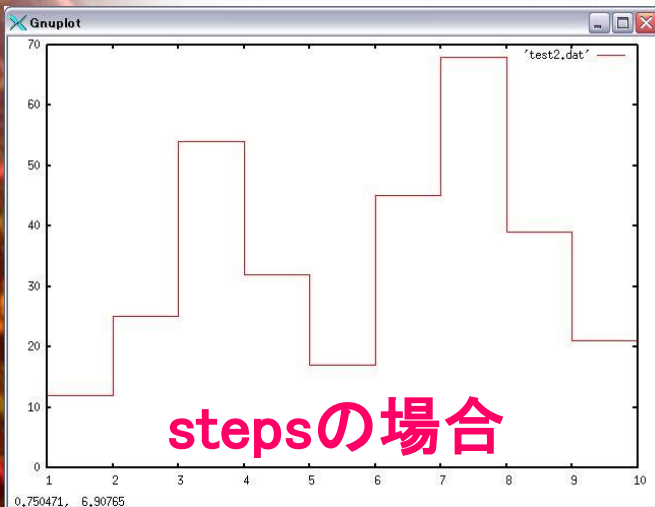
書式

- `gnuplot> plot 'ファイル名' with steps` : (x,y)から始まるヒストグラム
- `gnuplot> plot 'ファイル名' with fsteps` : (x,y)で終わるヒストグラム
- `gnuplot> plot 'ファイル名' with histeps` : (x,y)を中心とするヒストグラム

例

- `gnuplot> plot 'test2.dat' with steps`
- `gnuplot> plot 'test2.dat' with fsteps`
- `gnuplot> plot 'test2.dat' with histeps`
- test2.datは以下にある

`/home/teacher/z6wt01in/SAMPLE/test2.dat`



誤差グラフ(測定値に誤差がある場合)

- データの書式: $(x, y \pm \delta y)$ の場合

- 1行に

$x \ y \ \delta y$

がスペースで区切られたものを用意

- 書式

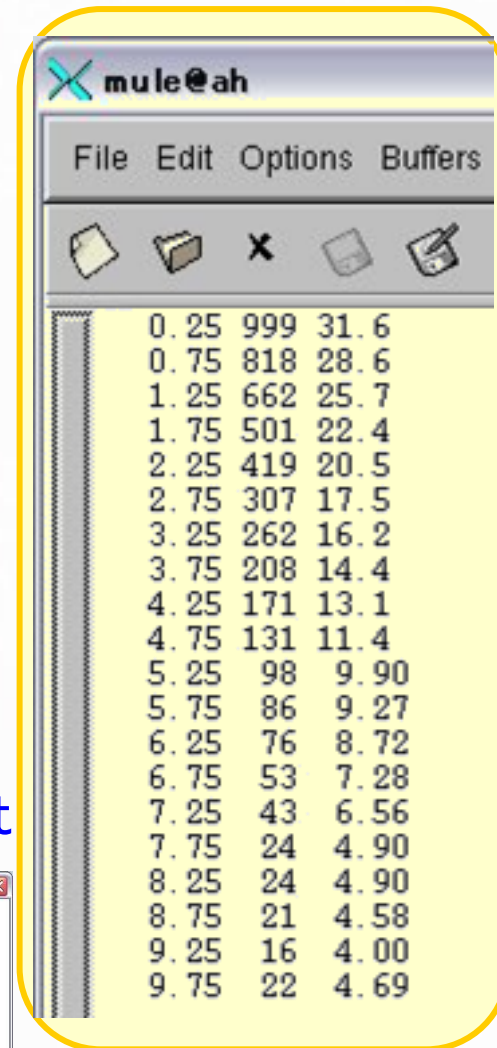
- gnuplot> plot 'ファイル名' with yerrorbars

- 例

- gnuplot> plot 'mu_decay.dat' with yerrorbars

- mu_decay.datは以下にある

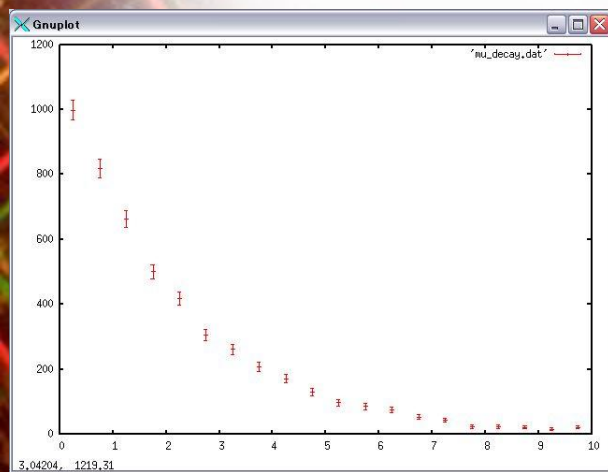
/home/teacher/z6wt01in/SAMPLE/mu_decay.dat



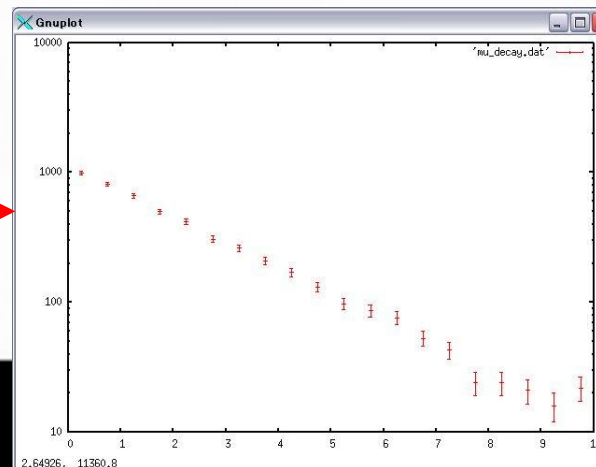
mule@ah

File Edit Options Buffers

0.25	999	31.6
0.75	818	28.6
1.25	662	25.7
1.75	501	22.4
2.25	419	20.5
2.75	307	17.5
3.25	262	16.2
3.75	208	14.4
4.25	171	13.1
4.75	131	11.4
5.25	98	9.90
5.75	86	9.27
6.25	76	8.72
6.75	53	7.28
7.25	43	6.56
7.75	24	4.90
8.25	24	4.90
8.75	21	4.58
9.25	16	4.00
9.75	22	4.69



set logscale y



最小二乗法

有効な場合

- 確率密度関数が不明(最尤法が適用できない)
- 独立変数 x_i に対して測定値 y_i が誤差 σ_i を伴って得られている

簡単のため、 k 次の多項式

$$f(x) = \sum_{j=0}^k a_j x^j$$

をデータに当てはめる(データを最も良く再現する a_j を求める)ことを考える。最小二乗法では、関数 $f(x)$ とデータ y_i とのズレの度合いを、

$$\chi^2 \equiv \sum_{i=1}^n \frac{(y_i - f(x_i))^2}{\sigma_i^2}$$

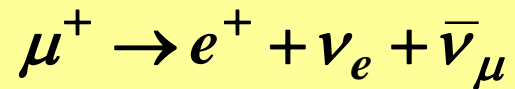
で表し、この χ^2 を係数 a_j を調整して最小にする。したがって、各 a_j に対する偏微分が0になるような a_j の組を求めれば良いので、解くべき問題は

$$\frac{\partial \chi^2}{\partial a_j} = \sum_{i=1}^n \frac{2(y_i - f(x_i))}{\sigma_i^2} x_i^j = 0 \quad (j = 0, \dots, k)$$

で与えられる $(k+1)$ 元連立1次方程式を解くことに帰着する。

μ 粒子の崩壊 (最小二乗法の例)

- ・ μ 粒子は弱い相互作用により、ある寿命 τ で、



と崩壊する。

- ・ μ 粒子の崩壊が指数分布

$$f(t; \tau) = \frac{1}{\tau} e^{-t/\tau} = \lambda e^{-\lambda t} \quad (\lambda \equiv 1/\tau)$$

に従うとして、データから寿命 τ を推定し、その誤差を検討する。

- ・ 例: μ 粒子の崩壊数を $0.5\mu\text{ s}$ 毎に測定して、右表のようなデータを得た場合

時間 (μs)	崩解数	時間 (μs)	崩解数
0.25	999	5.25	98
0.75	818	5.75	86
1.25	662	6.25	76
1.75	501	6.75	53
2.25	419	7.25	43
2.75	307	7.75	24
3.25	262	8.25	24
3.75	208	8.75	21
4.25	171	9.25	16
4.75	131	9.75	22

Gnuplotによる最小二乗フィット

- ・ Gnuplotには、非線形関数の最小二乗法が組み込まれている
 - プログラムを組まなくても手軽に使える
 - **結果をすぐに視覚化できる**

- ・ 手順(μ 粒子の寿命測定(決定)を例に)

- データファイルを用意
(例えばx,y,dyの並び)
- 関数を定義

```
gnuplot> f(x)=a*exp(-b*x) ↵
```

- 変数の初期値を設定(正しく収束させるために必要)

```
gnuplot> a=1000 ↵
```

```
gnuplot> b=0.5 ↵
```

- 実行

```
gnuplot> fit f(x) 'mu_decay.dat' using 1:2:3 via a,b ↵
```

データの並びが x,y, δy である

変数a,bを最適化する



Gnuplotによる最小二乗フィット(続き)

手順(続き)

- 結果

Final set of parameters

Asymptotic Standard Error

=====

=====

a = 1137.76

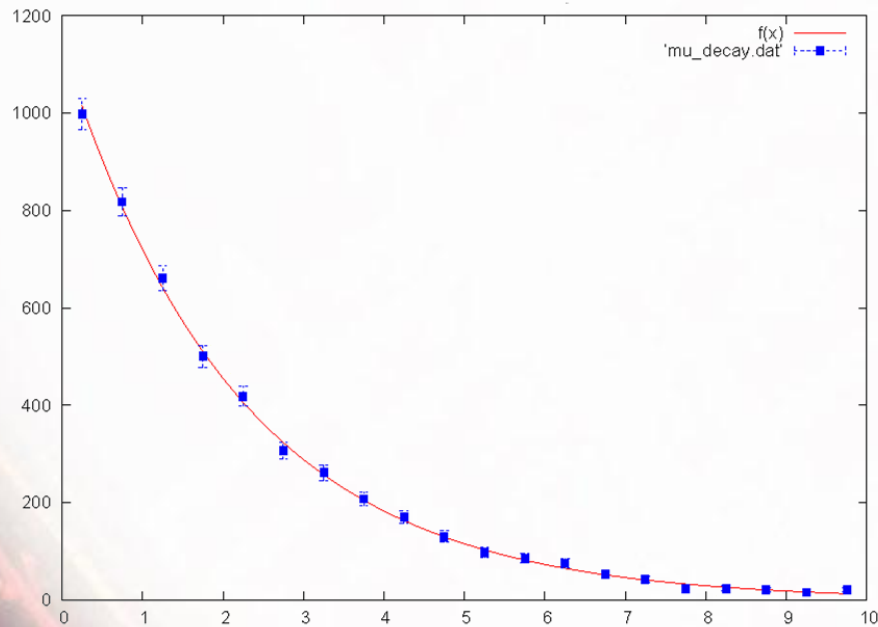
+/- 19.49 (1.713%)

b = 0.455834

+/- 0.006014 (1.319%)

- 結果の表示

```
gnuplot> plot f(x), 'mu_decay.dat' with yerrorbars 3 5
```



$$\tau = \frac{1}{b} = 2.19 \pm 0.03 \text{ } (\mu\text{s})$$

