可視化の基礎演習

担当教員　大野

概要

- 1次元データの可視化
  - データのフォーマット
  - プロット
  - 線種などの変更
  - グラフや軸の名前など

参考文献
川原稔：“gnuplotパーフェクト・マニュアル”，ソフトバンクパブリッシング株式会社

コマンドについて

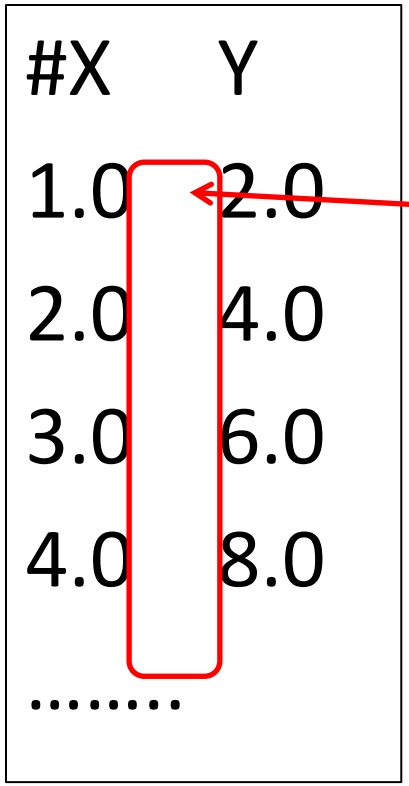
- set␣xxx⏎とすると、unset␣xxx⏎あるいはreset⏎するまで、その設定は有効
- コマンドが長くなったら ¥ を入れて改行できる

Example

>plot␣cos(x), sin(x), ¥⏎

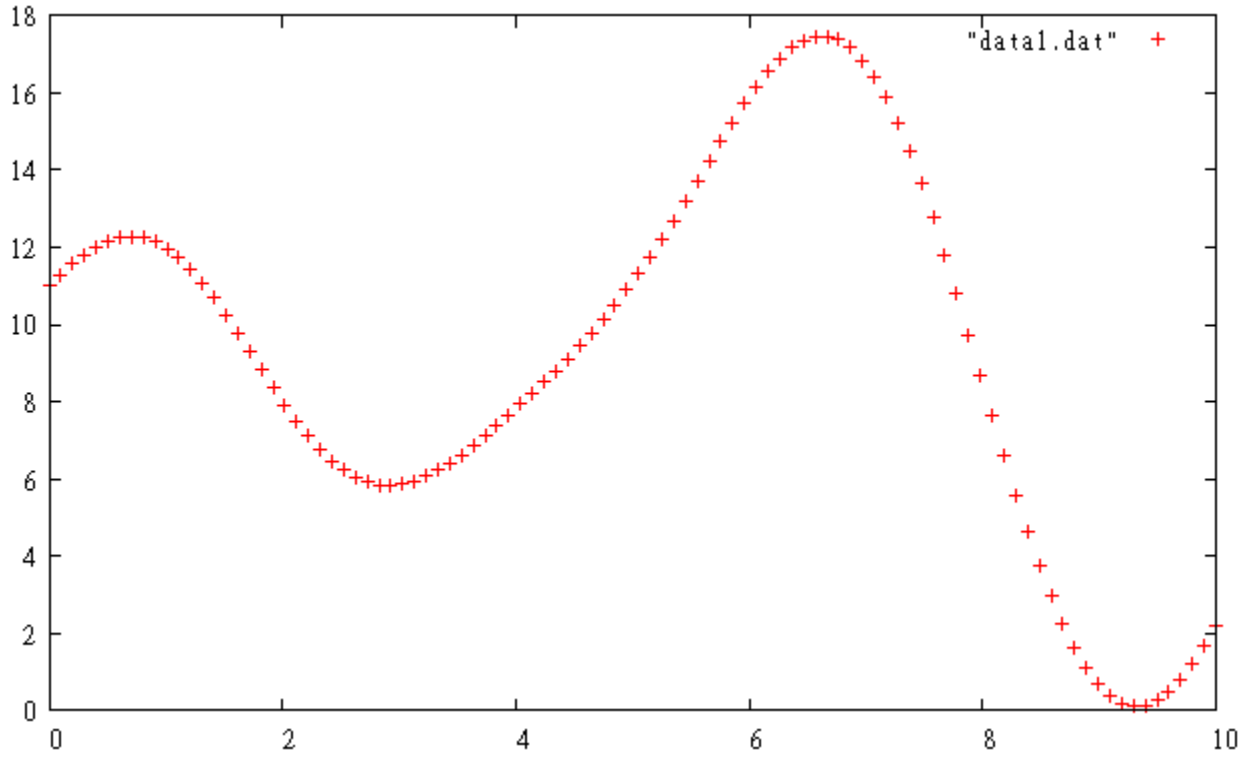
> tan(x)⏎

ファイルフォーマット -1



- テキストファイル
- スペースやタブで区切られている

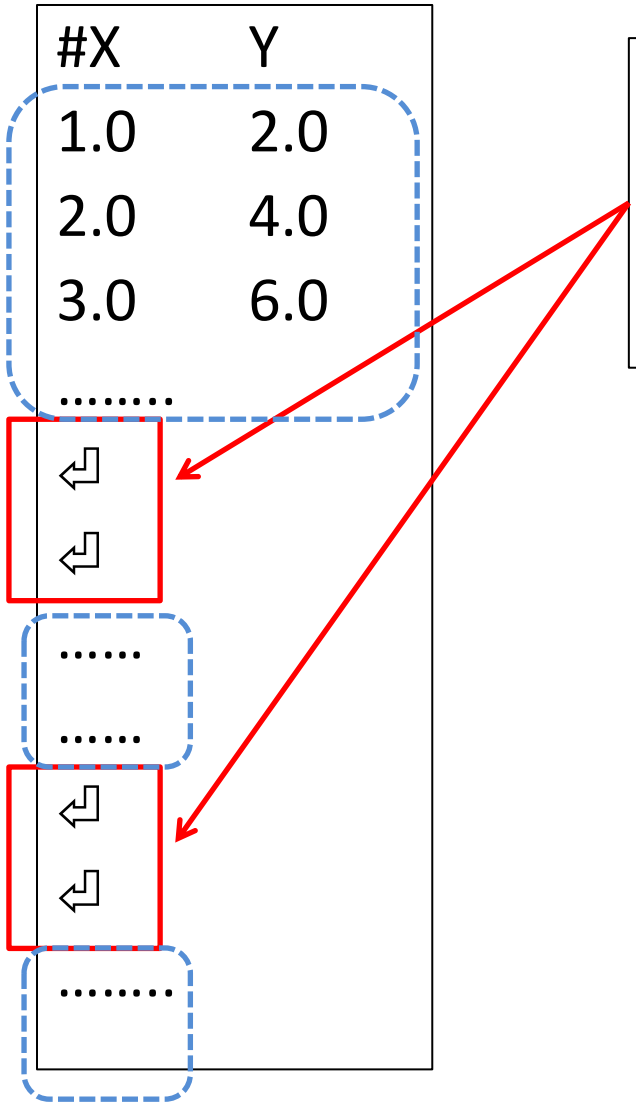
ファイルフォーマット -2



data1.datをプロットしてみよう

gnuplot> plot␣“data1.dat”⏎

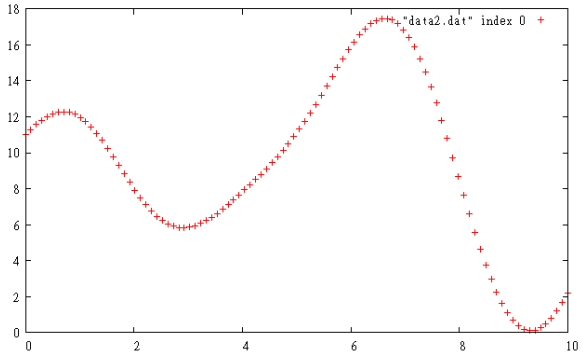
ファイルフォーマット -3



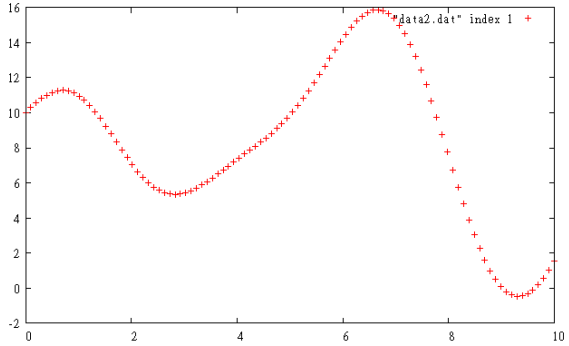
いくつかのデータを改行×2で区切って同じファイルに入れることもできる
初めのブロックから、index 0, index 1,…としてアクセスできる.

ファイルフォーマット -4

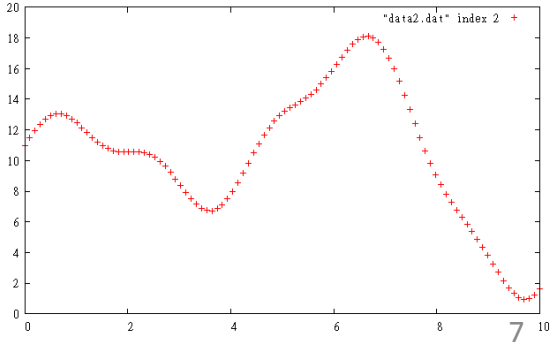
gnuplot> plot␣"data2.dat"␣index␣0⏎



gnuplot> plot␣"data2.dat"␣index␣1⏎



gnuplot> plot␣"data2.dat"␣index␣2⏎

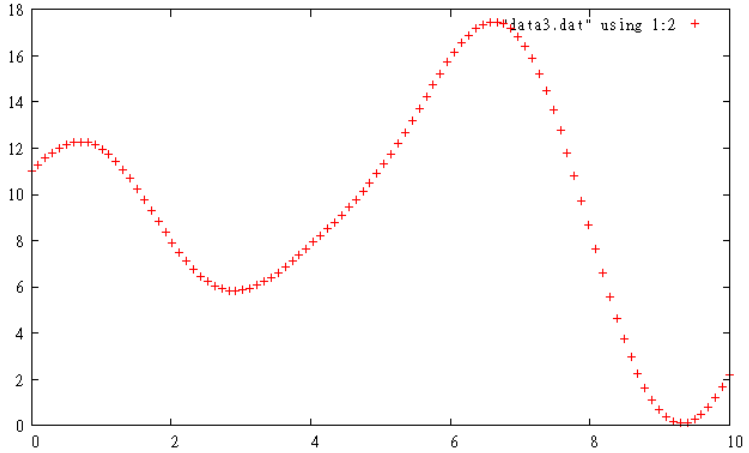


似てるけど、、、

ファイルフォーマット -5

- 2カラム以上でもよい。

#x	y1	y2	y3	….
0.0	1.0	1.2	0.9	…
1.0	1.2	1.4	0.8	
2.0	1.5	1.3	0.5	
…….				

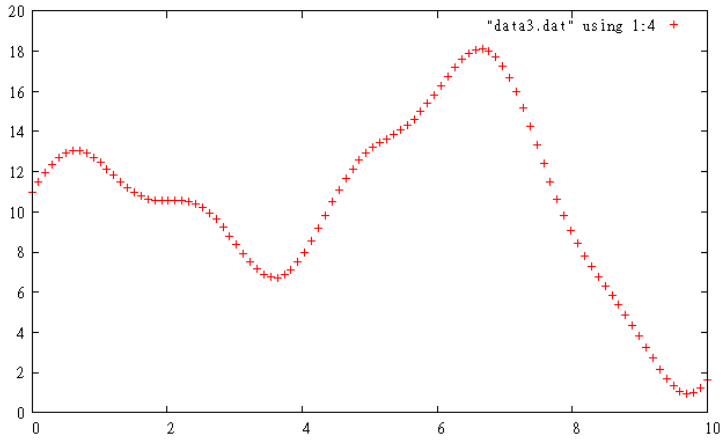


gnuplot> plot␣"data3.dat"␣using␣1:2
⏎ または (or)
gnuplot> plot␣"data3.dat"⏎

ファイルフォーマット -5A

- 2カラム以上

#x	y1	y2	y3	….
0.0	1.0	1.2	0.9	…
1.0	1.2	1.4	0.8	
2.0	1.5	1.3	0.5	
……				




```
gnuplot> plot␣"data3.dat"␣using␣1:4⏎
```

ファイルフォーマット -6

plot [範囲指定] "File" [index] [using] [smooth] [with スタイル]

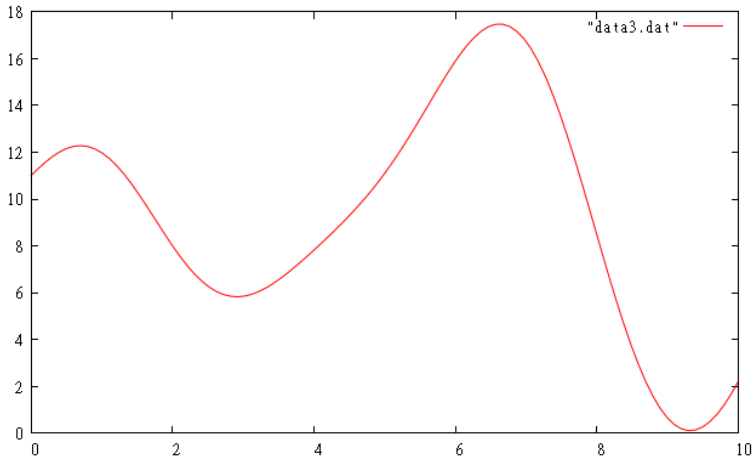
順番が違うとエラーが出る(かも)

プロット -1

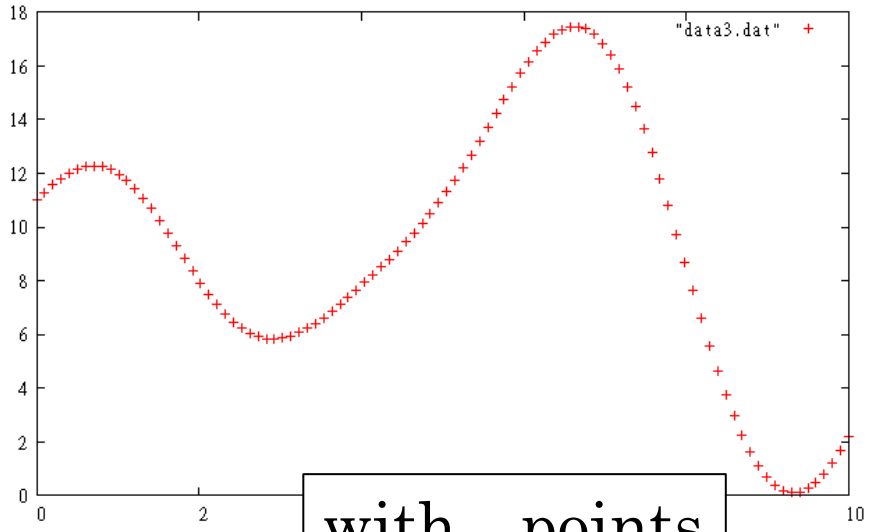
- デフォルトでは、”+”などのポイントのみのプロット
- 線、線とポイント、棒グラフ等々でグラフを描くことができる

例えば

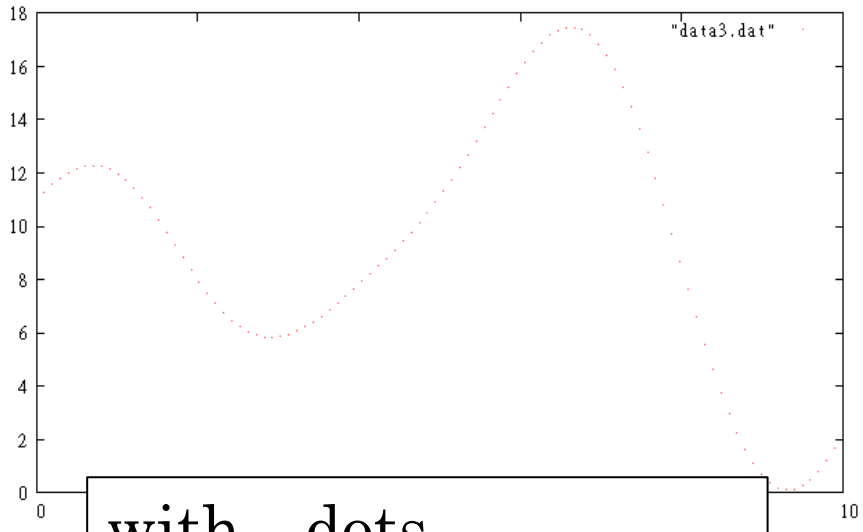
gnuplot> plot␣"data3.dat"␣with␣lines⏎



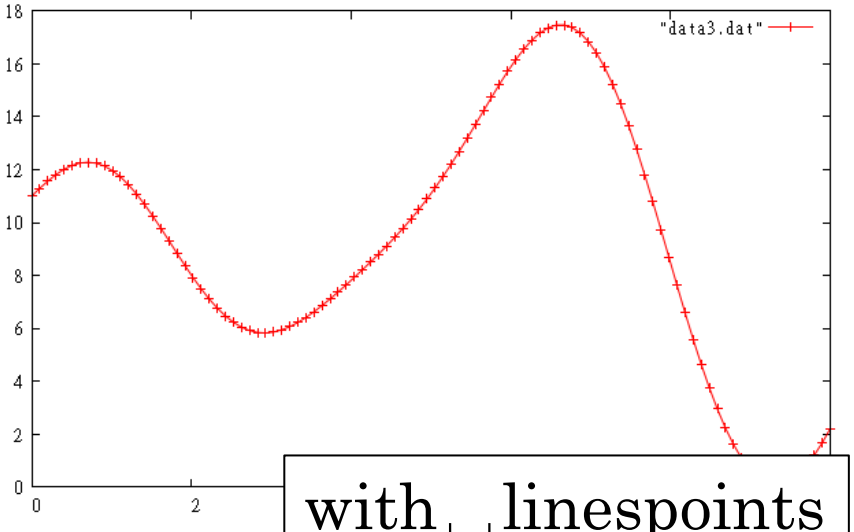
プロット -2



with␣points

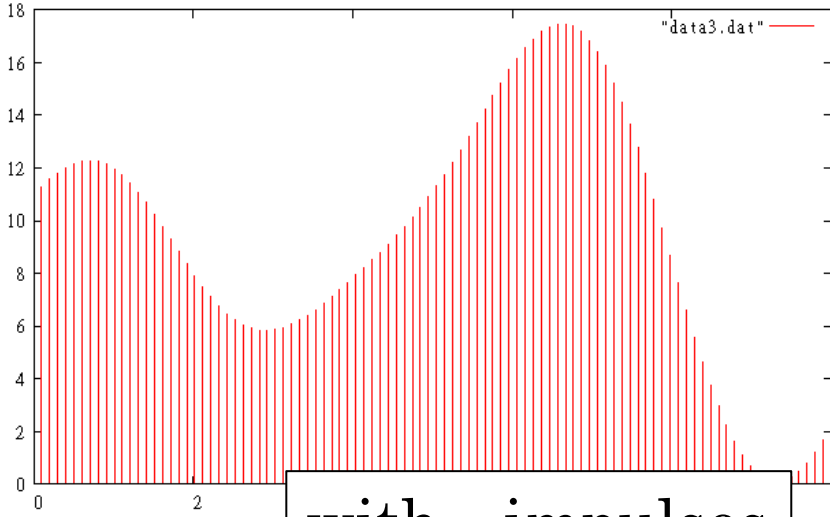


with␣dots
よく見えないけど点が
うってある

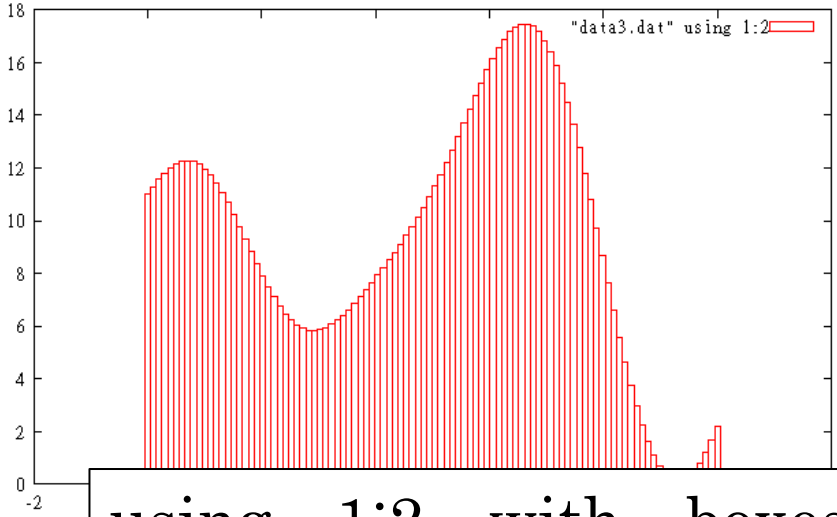


with␣linespoints

プロット -3



with␣impulses



using␣1:2␣with␣boxes
3カラム以上あると、3カラム目の値が棒グラフの幅になる

プロット -4

Exercise 1

- data3.dat を lines, linespoints, dots, impulses, boxesで描いてみよう

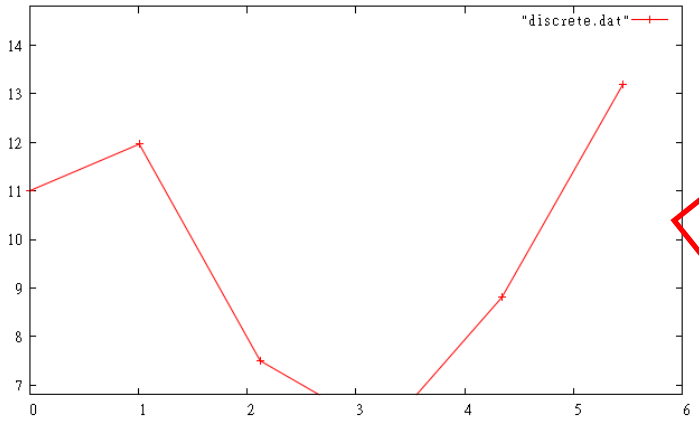
- steps, fsteps, histepsでは、どのようなグラフができるか？

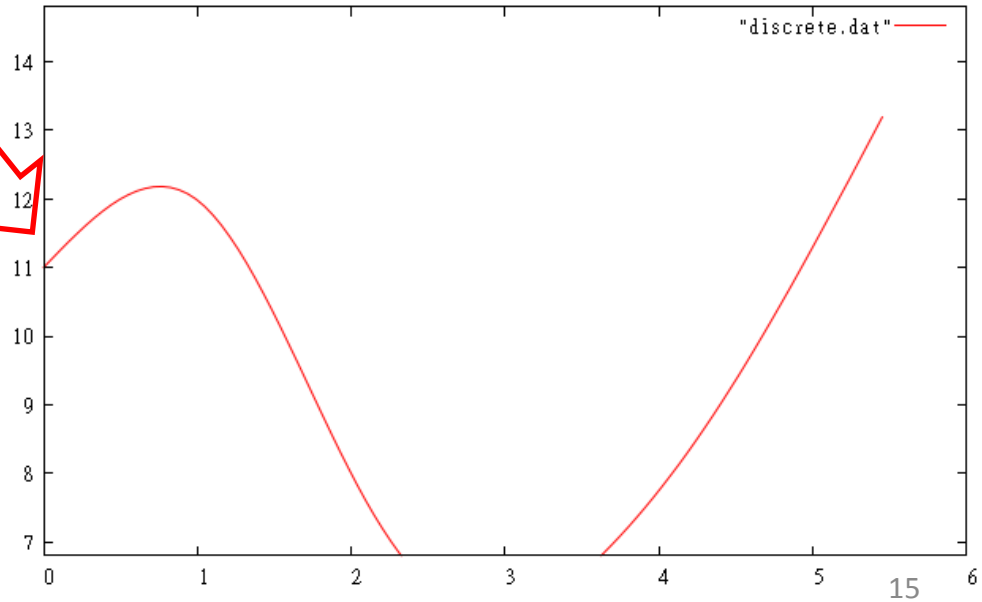
プロット -5

基本的に折れ線で、結んでいる

gnuplot>plot␣"discrete.dat"␣with␣linespoints⏎

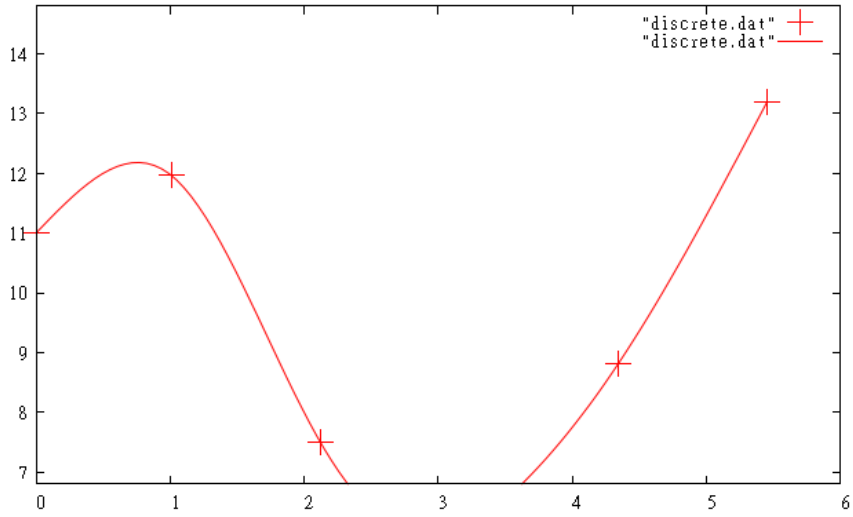
gnuplot>plot␣"discrete.dat"␣smooth␣cspline⏎



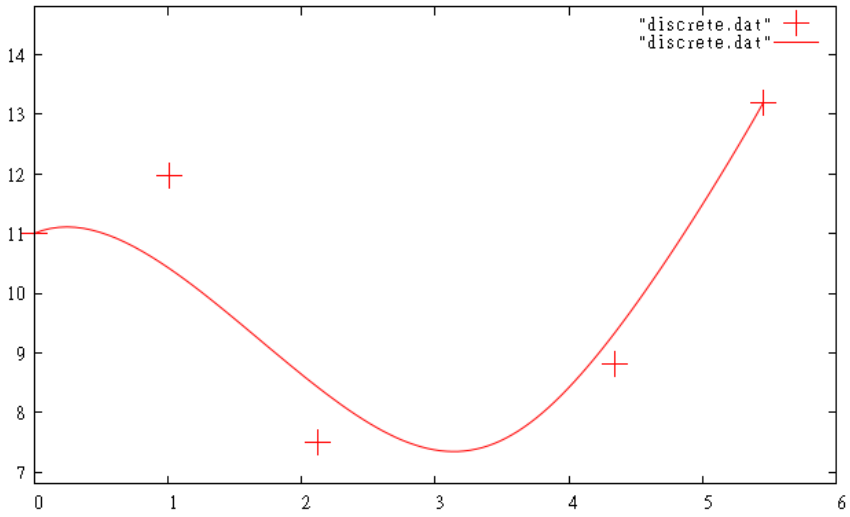


プロット -6

gnuplot>plot␣"discrete.dat"␣smooth␣cspline⏎
gnuplot>plot␣"discrete.dat"␣smooth␣bezier⏎



cspline



bezier

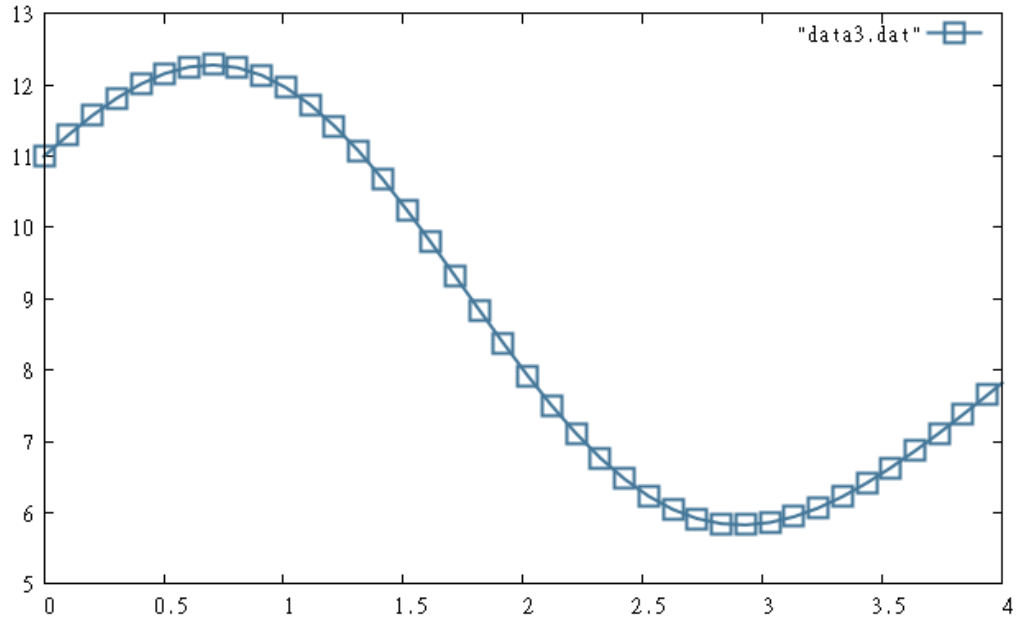
線種などの変更 -1

- ポイントや線の幅などを変更できる

Line		Point	
linewidth	線の幅	pointsize	Pointの大きさ
linetype	線の種類	pointtype	Pointの種類
linecolor	線の色 rgbcolorなどで指定		

線種などの変更 -2

Example 1



gnuplot>plot␣"data3.dat"␣with␣linespoints␣linewidth␣2␣
pointsize␣2␣pointtype␣4␣linecolor␣rgbcolor␣"#447799"⏎

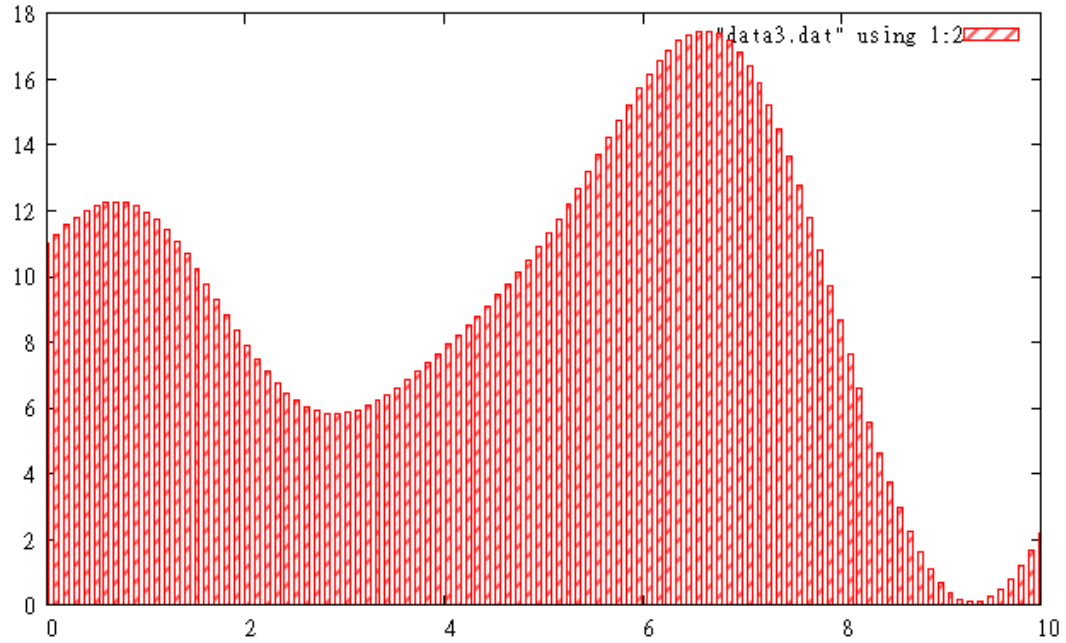
線種などの変更 -3

- 棒グラフ

box	
boxwidth	幅 set boxwidth 0.5 relative set boxwidth 2
style fill solid	set style fill solid 1.0 “1.0” is density (密度)
style fill pattern	set style fill pattern 5 “5” is the pattern number 5はパターンの番号

線種などの変更 -4

Example 2



gnuplot> set␣boxwidth␣0.5␣relative⏎

gnuplot> set␣style␣fill␣pattern␣5⏎

gnuplot> plot␣"data3.dat"␣using␣1:2␣with␣boxes⏎

線種などの変更 -5

- linetype, pointtype, patternは

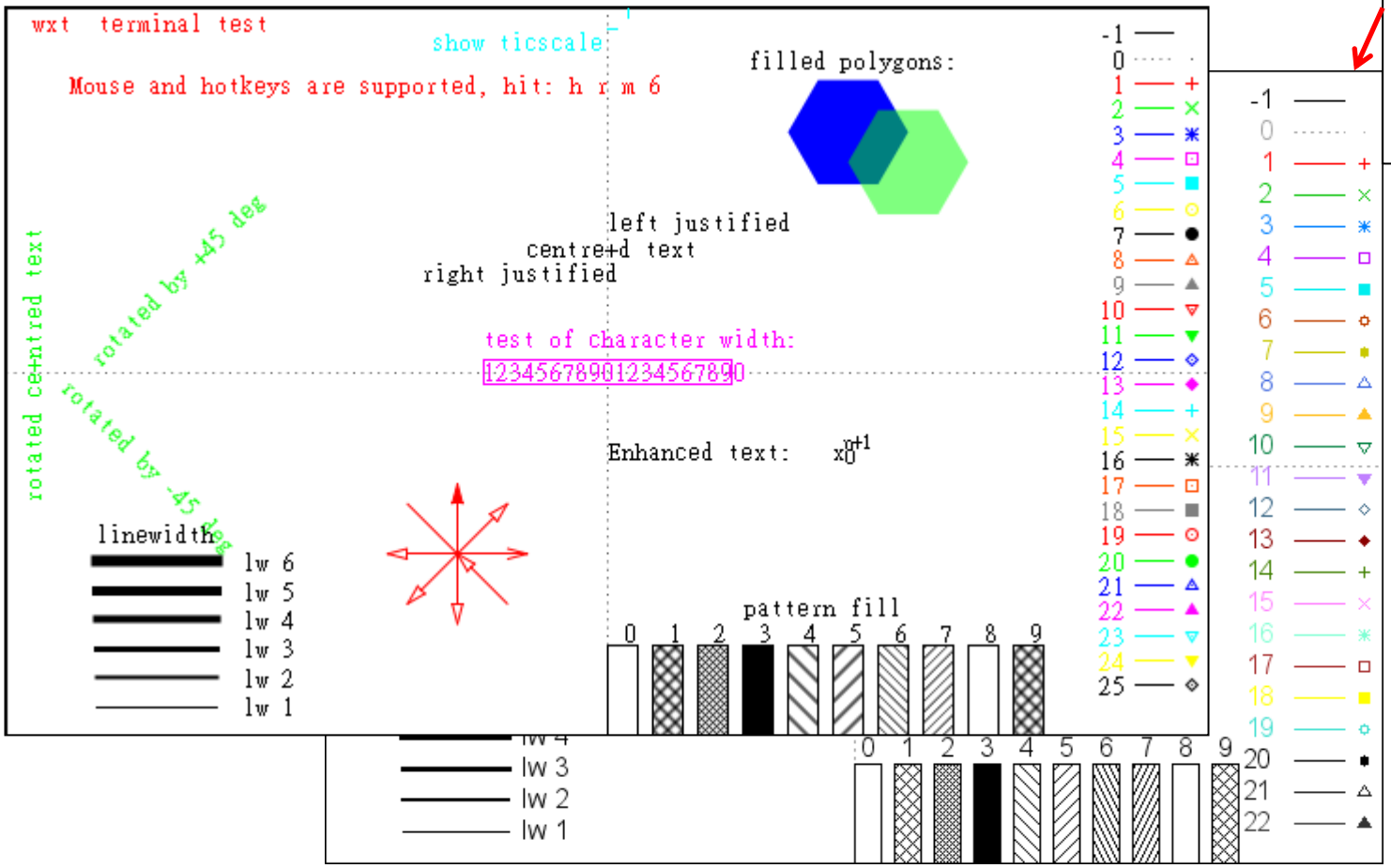
gnuplot>test⏎

で確認できる

- x11, wxt, eps, png, ….では、番号が同じでも、色や形が違う場合があるので注意

線種などの変更 -6

- wxt のテスト結果



Png
色など少し違う

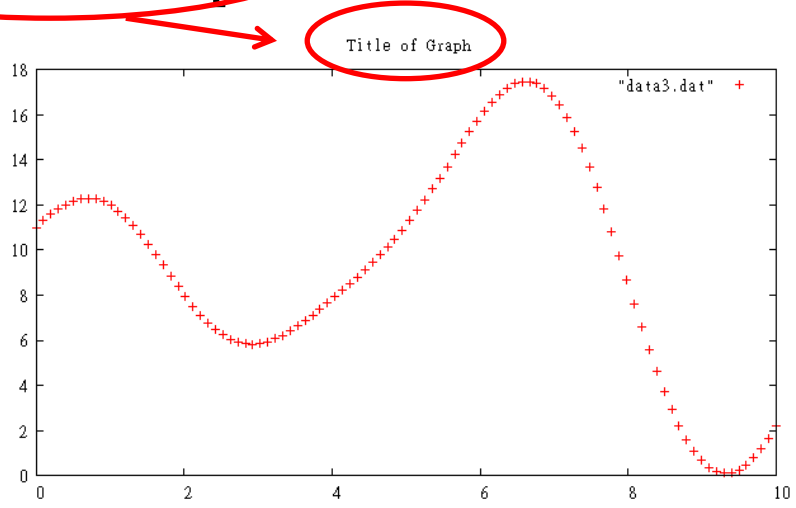
グラフや軸の名前など -1

- Excelのように、グラフや軸のタイトルを入れたい

1. Title of Graph

gnuplot> set␣title␣"Title of Graph "⏎

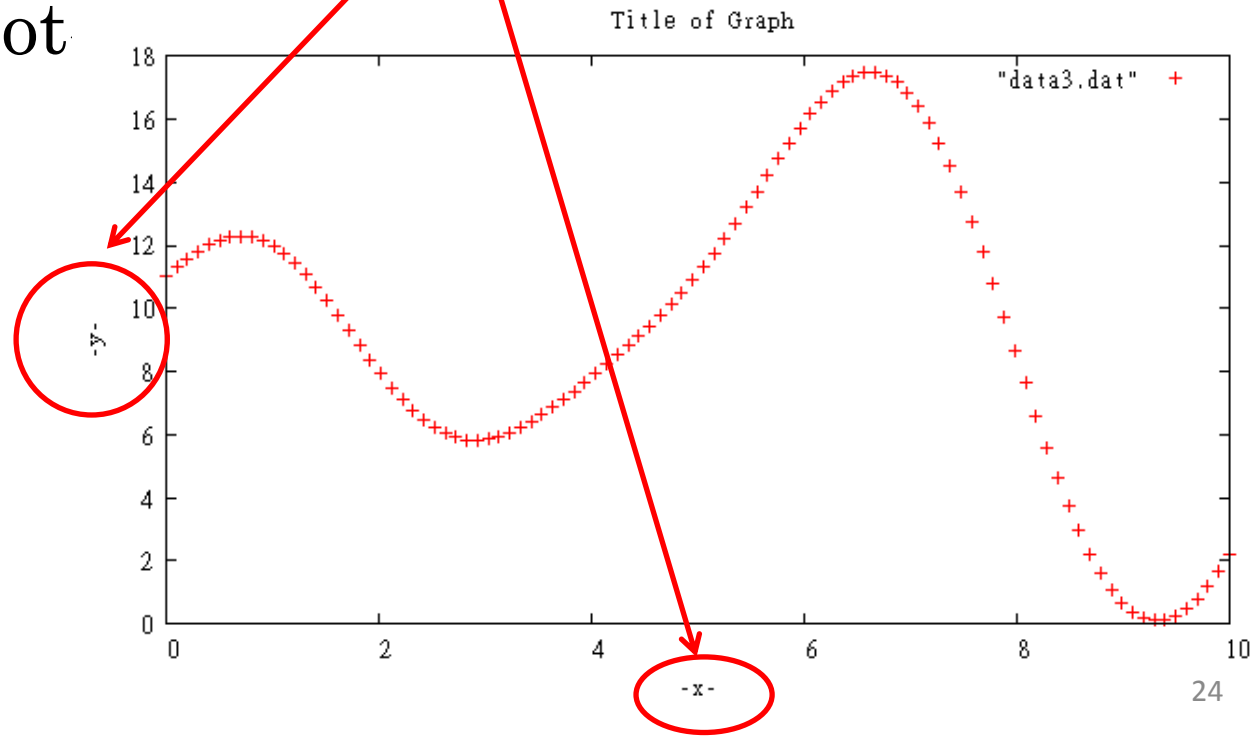
gnuplot> plot␣"data3.dat"



グラフや軸の名前など -2

- X,Y軸の名前

```
gnuplot> set␣xlabel␣"-x-"↵
gnuplot> set␣ylabel␣"-y-"↵
gnuplot> replot
```



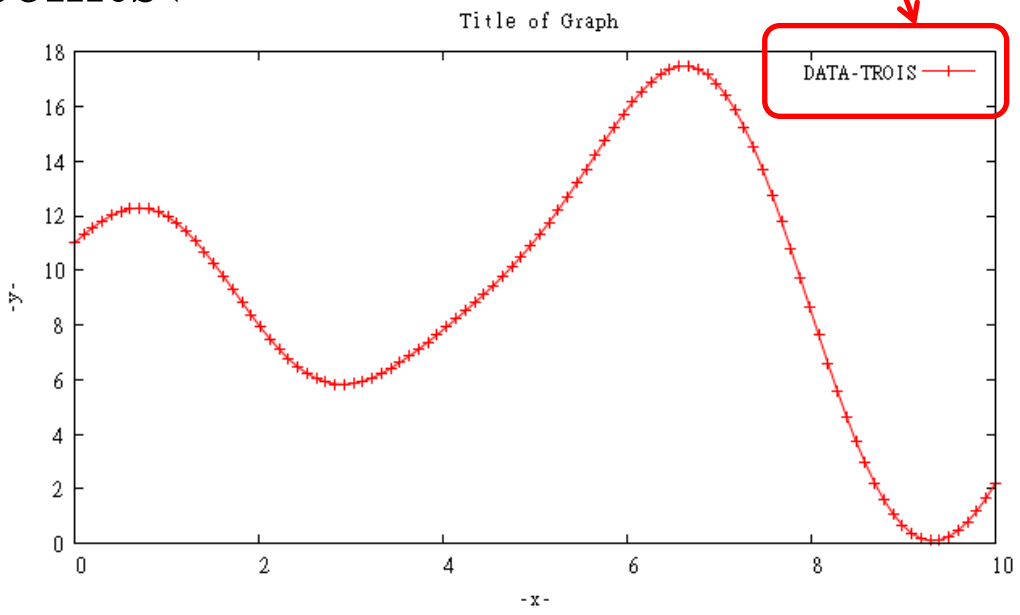
グラフや軸の名前など -3

- 凡例の名前

gnuplot> set␣xlabel␣"-x-"⏎

gnuplot> set␣ylabel␣"-y-"⏎

gnuplot> plot␣"data3.dat" ␣title ␣“DATA-TROIS”
␣with␣linespoints⏎



グラフや軸の名前など -4

- 凡例の位置やタイトル

gnuplot> unset␣key⏎ :凡例を消す

gnuplot> set␣key␣title␣"title of legend"⏎ # 凡例のタイトル

gnuplot> set␣key␣<POSITION>⏎

POSITION	
top	上
bottom	下
left	左
right	右
outside	グラフ外、右
below	グラフ外、下

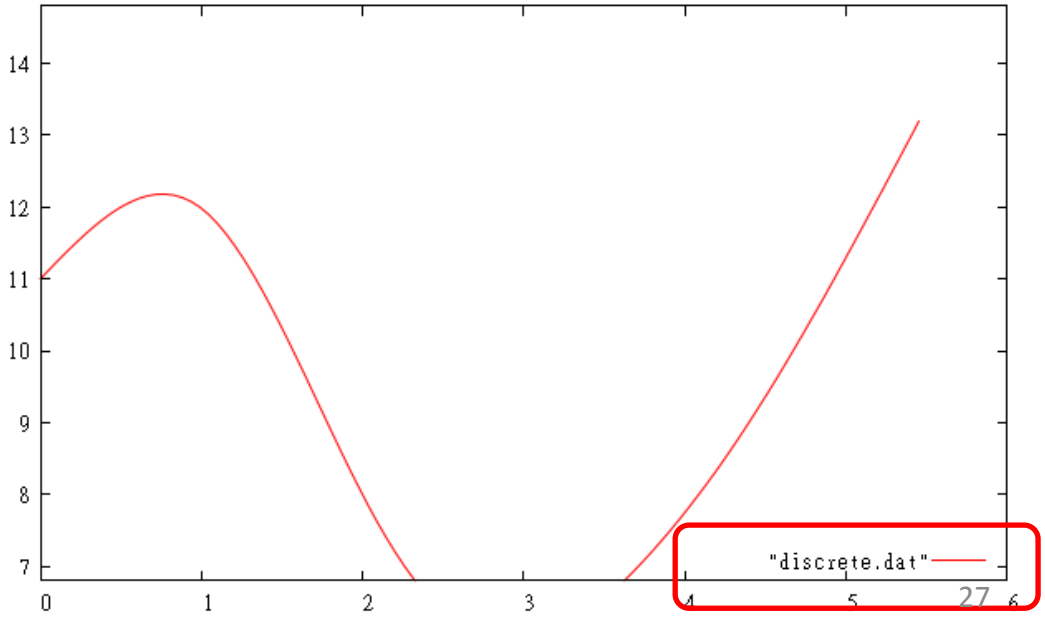
グラフや軸の名前など -5

Example 3

gnuplot>set␣key␣bottom⏎

gnuplot>set␣key␣right⏎

gnuplot>plot␣"discrete.dat"␣smooth␣cspline⏎



グラフや軸の名前など -3

Exercise 2

Slide 23-27までを実行してみよう

凡例の位置を変えてみよう

Appendix -1

線の種類などは

plot␣"data.dat"␣with␣linespoints␣1␣3⏎

plot␣"data.dat"␣with␣points␣6⏎

などのようにも指定できる

with=w, linespoints=lpなどの略記もある

例:plot␣cos(x)␣w␣lp⏎ =
plot␣cos(x)␣with␣linespoints⏎

Appendix -2
拡張テキストモード -1

- ラベルなどをもっとよくしたい
- ギリシャ文字やTeXのようなコマンドが使える
- 拡張モードにするには、

gnuplot> set␣termoption␣enhanced⏎(set current terminal on enhanced mode)

or

gnuplot> set␣terminal␣wxt(png, postscript eps…)␣enhanced⏎

Appendix -2
拡張テキストモード -2

a^x or a^{x}: a^x

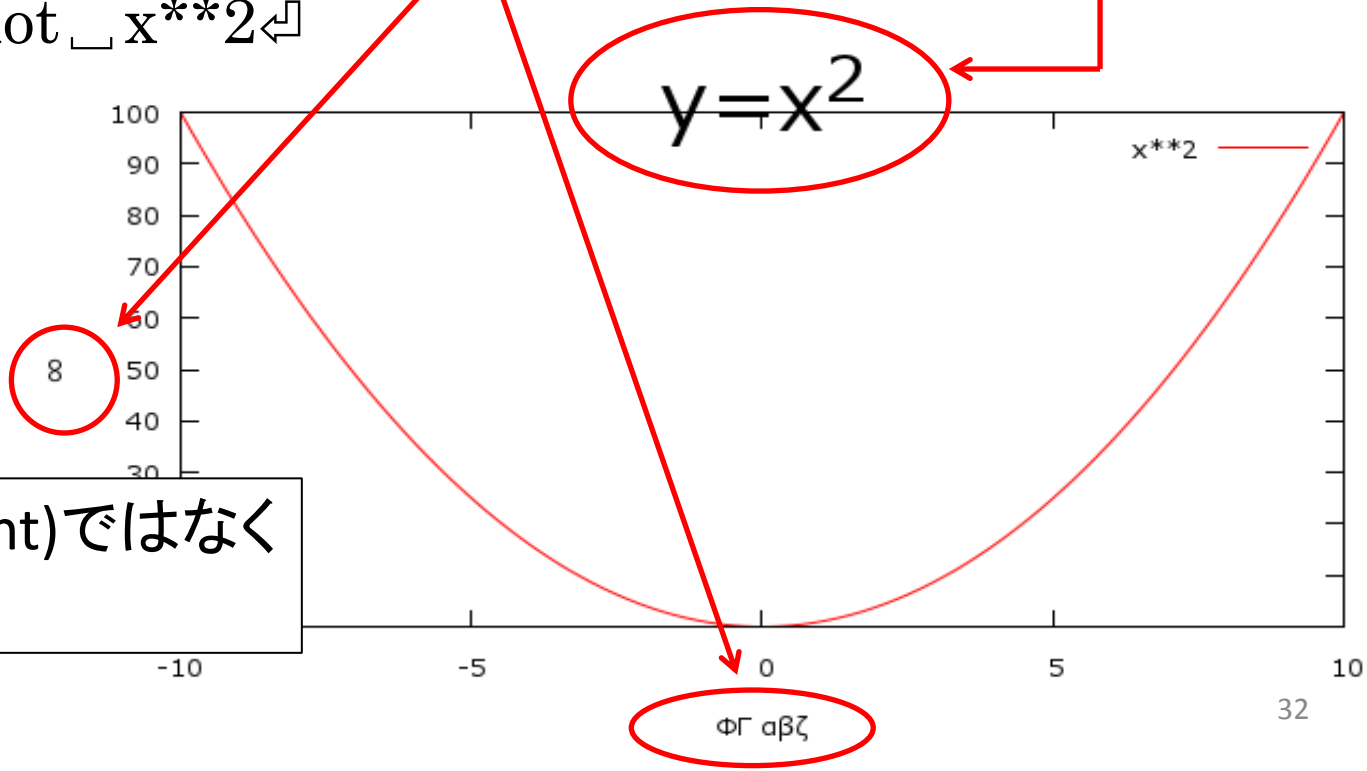
a_x or a_{x}: a_x

{/font } : フォントの指定, example: {/Times Title}
{/=X}: フォントの大きさの指定　X = size of font
{/Symbol }: ギリシャ文字 Greek : {/Symbol a} -> α

{/Times=30}: size=30のfont=Times(同時に指定)

Appendix -2
拡張テキストモード -3

```
gnuplot> set␣termoption␣enhanced⏎
gnuplot> set␣title␣"{/=30 ␣y=x^2}"⏎
gnuplot> set␣xlabel␣"{/Symbol␣FG␣abz}"⏎
gnuplot> set␣ylabel␣“{/Symbol␣¥245}"⏎
gnuplot> plot␣x**2⏎
```



八(はち，eight)ではなく
無限大

Appendix -3
アニメーション -1

- gnuplotのスクリプトを用意することで実現できる
- 時系列データの可視化に便利
- Example:
  - animation.plt
    - アニメーション用のスクリプト
  - data4.dat
    - データ。改行x2で区切られたデータ(30個分)が入っている。

Appendix -3
アニメーション -2

- スクリプトの簡単な解説
  - if 文:　条件分岐　C言語のifと似ている

&&	And	!=	Not equal
||	Or	<, <=	＜, ≦
==	equal	>, >=	＞, ≧

  - pause x: x秒停止(wait for x seconds)。-1としたときは、エンターキー待ち
  - reread: スクリプトをもう一度読み込む。変数は初期化されないことに注意
  - sprintf: 文字列に置き換わる。C言語のsprintfに似ている
  - exists("X")　変数Xが定義されているか否か
  - load␣"animation.plt"⏎でスクリプトを実行。

Appendix -3
アニメーション -3

- animation.pltを少し改造すると、連番画像作成スクリプトになる
- img_sequence.plt:　表示をpng出力に変えた
- load␣"img_sequence.plt"⏎ で実行
- 連番画像は、ImageJ, Quick Time Pro, Adobe製品でムービーにできる (第3回)