

# 最低限グラフィックス

## gnuplot

# はじめに

- 今後履修することになる実験や実習では, 自分で測った/観測した, あるいは自分で計算したたくさんのデータのグラフを描く機会が増えるだろう.
- もちろん, 卒業研究発表・卒業論文にはたくさんのグラフを載せることになる.
- 今回は, **Unix/Linux** で広く使われているグラフ作成ソフトウェアである **gnuplot** を使うことにする.

# gnuplot とは

- Unix/Linux, Windows, Mac OSX 等で使うことのできる, コマンドラインで実行する作図ソフトウェア.
- 2次元の図だけでなく, 3次元の図を描くことも可能.
- 対話的に図を描くことも, 非対話的に利用することも可能.
- フリーソフトウェア

# “gnuplot” の発音

- 人によって様々（少なくとも日本人は）.
  - 「グヌープロット」 「ニュープロット」 「ジーエヌユープロット」
    - 外国人の発音は「ニュープロット」？
- 補足
  - “GNU” は “GNU’s Not Unix!” の頭字語.
    - “GNU” の発音は [ˈɡnuː] （グヌー（？））
      - <https://www.gnu.org/gnu/pronunciation.ja.html>
  - しかし, “gnuplot is not related to the GNU project or the FSF in any but the most peripheral sense.” (<http://www.gnuplot.info/faq/faq.html#x1-70001.2>) であり, “GNU” の発音にこだわる理由はない.

# 起動方法

```
xxx@DESKTOP: $ gnuplot
```

← コマンドの後には [return]

```
GNUPLOT
```

```
Version 3.7 patchlevel 2 (+1.2.0rc2 2002/05/05)
```

```
last modified Sat Jan 19 15:23:37 GMT 2002
```

```
Copyright(C) 1986 - 1993, 1998 - 2002
```

```
Thomas Williams, Colin Kelley and many others
```

```
Type `help` to access the on-line reference manual
```

```
The gnuplot FAQ is available from
```

```
<http://www.ucc.ie/gnuplot/gnuplot-faq.html>
```

```
Terminal type set to 'aqua'
```

```
gnuplot>
```

← gnuplot のプロンプト; ここにコマンドを入力.

# 終了方法

```
xxx@DESKTOP: $ gnuplot
```

```
GNUPLOT
```

```
Version 3.7 patchlevel 2 (+1.2.0rc2 2002/05/05)
```

```
last modified Sat Jan 19 15:23:37 GMT 2002
```

```
Copyright(C) 1986 - 1993, 1998 - 2002
```

```
Thomas Williams, Colin Kelley and many others
```

```
Type `help` to access the on-line reference manual
```

```
The gnuplot FAQ is available from
```

```
<http://www.ucc.ie/gnuplot/gnuplot-faq.html>
```

```
Terminal type set to 'aqua'
```

```
gnuplot> quit ← 終了のコマンド
```

```
xxx@DESKTOP: $ ← シェルのプロンプト
```

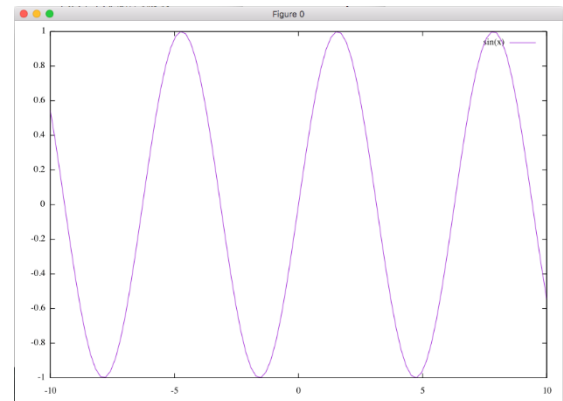
# 関数の描画 1

- gnuplot では, いくつかの関数を描画できる.
  - 三角関数, 指数関数, 対数, ...
- 例えば,

```
gnuplot> plot sin(x)
```

← コマンドの後には [return]

とすると, 右下図のようなウィンドウが表示される.



# 関数の描画 2

- 複数の関数を描画したい時には並べて書く.
- 例えば,

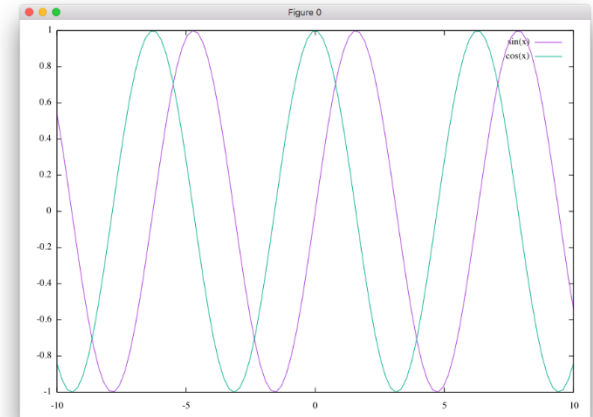
```
gnuplot> plot sin(x), cos(x)
```

← コマンドの後には [return]



カンマ

とすると, sine と cosine が重ね書きされる.





# 関数の描画 3

- 今後, 授業や研究の過程で様々な関数を目にするでしょう. それは簡単な関数かもしれませんが, 複雑な関数かもしれません.
- それらの関数の振る舞いを把握する上でグラフを描くことはとても有効.
- 振る舞いのわからない関数に出会った時には `gnuplot` でグラフにしてみるとよいでしょう.

# ファイルに保存されたデータの描画 1

- **gnuplot** では, ファイルに収められたデータを描画できる.
- 例えば, ファイル“data.txt” に右のようにデータが収められているとする.

空白  
(何個でも良い)



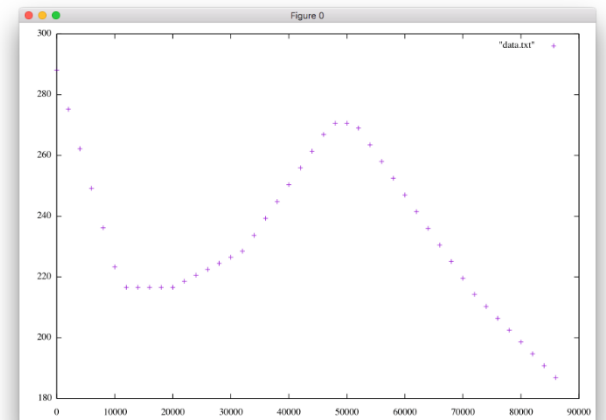
0.0	288.1
2.0	275.2
4.0	262.2
6.0	249.1
8.0	236.1
10.0	223.3
...	

# ファイルに保存されたデータの描画 2

- ファイル“data.txt”に収められたデータは下のようにして描画できる。

```
gnuplot> plot "data.txt"
```

このとき、データは点（シンボル）で描かれる。



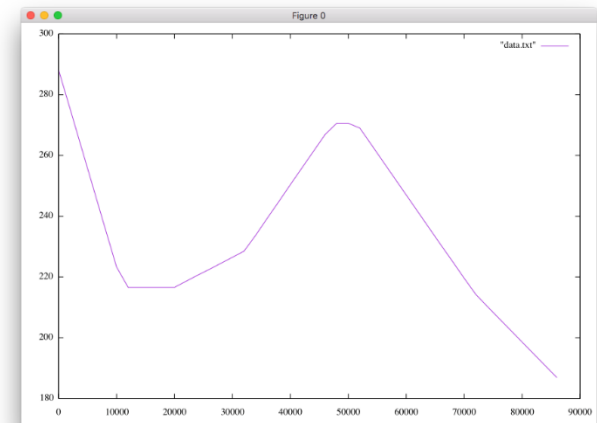
# ファイルに保存されたデータの描画 3

- さらに, 描画オプションを付けることで線の種類や太さなどを変えることができる.
- 例えば,

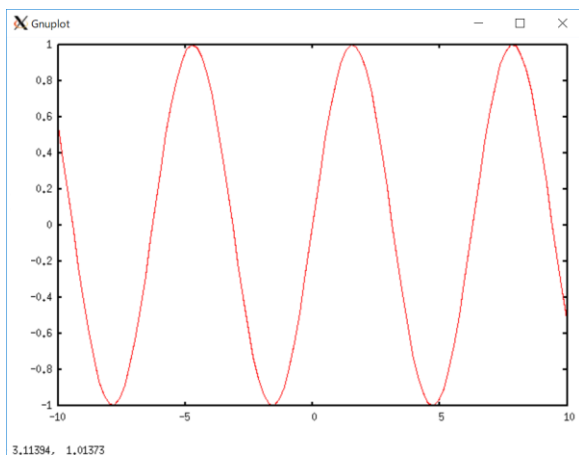
```
gnuplot> plot "data.txt" w l
```

とすると折れ線グラフになる.

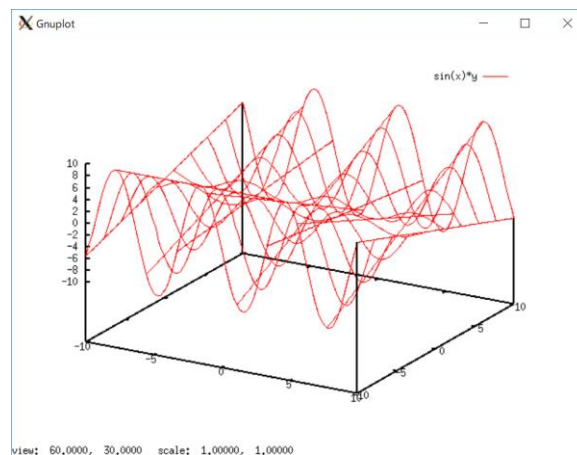
- ここで, “w l” は “with lines” の略.



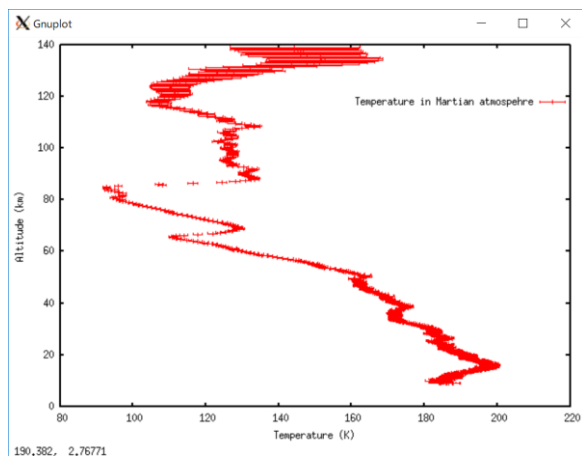
# gnuplot で作成したグラフの例



x の関数のグラフ



x, y の関数のグラフ



ファイルに保存されたデータのグラフ  
(観測データのグラフ)

# 実習

- 実習資料に基づいて gnuplot を使ってみましょう.