

情報システム演習 I

– Gnuplot 入門 –

森倉 悠介

y.morikura@thu.ac.jp

今日の授業

今日の目的

Gnuplot を (少し) 扱えるようになる。

今日の練習

- Gnuplot でグラフの描画を行ってみよう。

今日の授業

今日の目的

Gnuplot を（少し）扱えるようになる。

今日の練習

- Gnuplot でグラフの描画を行ってみよう。

Gnuplot について

Gnuplot とは？

1980 年代中頃からつくられはじめたグラフを描画するためのフリーのコマンドラインアプリケーション
Linux , Windows , MacOS に広く対応

プログラミングの結果：

結果が数字でしか得られないと、分析しにくい..

Gnuplot

プログラミングで得られた計算結果をグラフで表し人間にも理解しやすくできる。

Gnuplot について

Gnuplot とは？

1980年代中頃からつくられはじめたグラフを描画するためのフリーのコマンドラインアプリケーション
Linux, Windows, MacOS に広く対応

プログラミングの結果：

結果が数字でしか得られないと、分析しにくい..

Gnuplot

プログラミングで得られた計算結果をグラフで表し人間にも理解しやすくできる。

Gnuplot について

Gnuplot とは？

1980年代中頃からつくられはじめたグラフを描画するためのフリーのコマンドラインアプリケーション
Linux, Windows, MacOS に広く対応

プログラミングの結果：

結果が数字でしか得られないと、分析しにくい..

Gnuplot

プログラミングで得られた計算結果をグラフで表し人間にも理解しやすくできる。

Gnuplot

利点

- フリーソフト
- ほとんどの環境で使える
- 昔から使われているため多くの解説が Web にある
- 多くの画像形式に対応

欠点

- コマンド入力を行うため敷居が高い
- 使い方を理解すれば簡単に使えますのでこの機会に覚えましょう。

Gnuplot

利点

- フリーソフト
- ほとんどの環境で使える
- 昔から使われているため多くの解説が Web にある
- 多くの画像形式に対応

欠点

- コマンド入力を行うため敷居が高い
- 使い方を理解すれば簡単に使えますのでこの機会に覚えましょう。

Ubuntu へのインストール

基本的に，Linux 環境ではすでにインストールが行われているはずですが

確認：端末上で

```
$ gnuplot
```

反応がなければ，以下のインストールを行う．

インストール：端末上で

```
$ sudo apt install gnuplot
```

Gnuplot の起動

起動したいディレクトリに移動

起動：端末上で

```
$ gnuplot
```

を行うと起動し Gnuplot のコマンド入力モード

```
gnuplot>
```

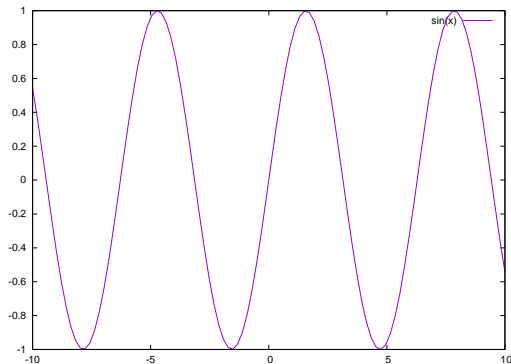
になります。

Gnuplot の利用

例えば：

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

とコマンドを打つと sin 関数が描画されます。



Gnuplot の終了

終了のコマンド：

```
gnuplot> exit
```

か

```
gnuplot> quit
```

とコマンドを打つと端末に戻ります。

画像挿入 !!

Gnuplot でのグラフの描画

- $y = \sin(x)$

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

- $y = 2x - 1$

```
gnuplot> plot 2*x-1
```

- $y = x^3 - x$

```
gnuplot> plot x**3-x
```

- $y = \tan(x) - x$

```
gnuplot> plot tan(x)-x
```

Gnuplot での2つのグラフの描画

- $y = \sin(x)$, $y = x$

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

- $y = 2x - 1$, $y = x^2$

```
gnuplot> plot 2*x-1,x*x
```

- $y = x^3 - x$, $y = x^4$

```
gnuplot> plot x**3-x, x**4
```

- $y = \tan(x) - x$, $y = \sin(x) - \cos(x)$

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

表示範囲の指定

範囲の指定

- $-1 < x < 1, y = 2x - 1$

```
gnuplot> plot [-1: 1] 2*x-1
```

- $-2\pi < x < 2\pi, y = \sin(x)$

```
gnuplot> plot [-2*pi: 2*pi] sin(x)
```

- $0 < x < 3, y = x^3 - x$

```
gnuplot> plot [0: 3] x**3-x
```

- $0 < x < 2\pi, y = \tan(x) - x$

```
gnuplot> plot [0: 2*pi] tan(x)-x
```

pi は円周率を表す .

表示範囲の指定

x 軸と y 軸の範囲の指定

- $-1 < x < 1, -5 < y < 5, y = 2x - 1$

```
gnuplot> plot [-1: 1][-5: 5] 2*x-1
```

- $-2\pi < x < 2\pi, -0.5 < y < 0.5 y = \sin(x)$

```
gnuplot> plot [-2*pi: 2*pi][-0.5: 0.5] sin(x)
```

- $-10 < y < 10 y = x^3 - x$

```
gnuplot> plot [[-10:10] x**3-x
```

- $0 < x < 2\pi, 0 < y < 2\pi, y = \tan(x) - x$

```
gnuplot> plot [0: 2*pi][0: 2*pi] tan(x)-x
```

pi は円周率を表す .

コマンドによる表示範囲の指定

set コマンド

- x 軸を $-1 < x < 1$ に設定する

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

- y 軸を $-0.5 < y < 0.5$ に設定する

```
gnuplot> set yrange [-0.5:0.5]
```

- x 軸の範囲を自動設定にする

```
gnuplot> set autoscale x
```

書き直し・重ね書き

replot コマンド

- 直前に描いたグラフを描き直す

```
gnuplot> replot
```

- 前のグラフに重ね描きする

```
gnuplot> plot x
```

```
gnuplot> replot sin(x)
```

3次元グラフ

splot コマンド

- $z = x + y$

```
gnuplot> splot x+y
```

- $z = x * x + y * y$

```
gnuplot> splot x*x+y*y
```

- $-1 < x < 1, 0 < y < 1, z = x * x + \sin(y)$

```
gnuplot> splot [-1:1][0:1] x*x+y*y
```

png 形式画像の保存

set コマンド

```
gnuplot> set terminal png
```

```
gnuplot> set output 'test.png'
```

```
gnuplot> replot
```

```
gnuplot> set terminal x11
```

eps 形式画像の保存

set コマンド

```
gnuplot> set terminal postscript eps
```

```
gnuplot> set output 'test.eps'
```

```
gnuplot> replot
```

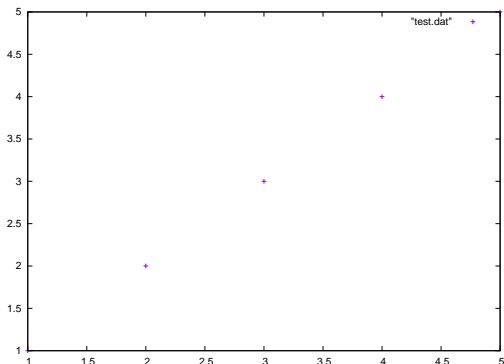
```
gnuplot> set terminal x11
```

データファイルからのプロット

以下のデータファイルのデータをプロット（描画）してみよう．

データファイル
(test.dat とする .)

```
#文字はコメント  
#x y の並び,  
#空白タブで区切る  
1 1  
2 2  
3 3  
4 4  
5 5
```



データファイルからのプロット

- test.dat のデータを点でプロットする .

```
gnuplot> plot "test.dat"
```

- test.dat のデータを直線でプロットする .

```
gnuplot> plot "test.dat" with linespoints
```

- test1.dat , test2.dat のデータを直線でプロットする .

```
gnuplot> plot "test1.dat" with linespoints, plot "test2.dat"  
with linespoints
```

- test.dat のデータをなめらかな曲線でプロットする .

```
gnuplot> plot "test1.dat" with linespoints
```

グラフ表示の設定

- グラフにタイトルをつける

```
gnuplot> set title "test"
```

- x 軸にラベルをつける

```
gnuplot> set xlabel "x"
```

- y 軸にラベルをつける

```
gnuplot> set ylabel "y"
```

- x 軸のメモリを変更する（1 から 5 まで，0.5 刻み）

```
gnuplot> set xtics 1, 0.5, 5
```

- グリッド（格子）を表示する

```
gnuplot> set grid
```


グラフ表示の設定

- グラフにタイトルをつける

```
gnuplot> set title "test"
```

- x 軸にラベルをつける

```
gnuplot> set xlabel "x"
```

- y 軸にラベルをつける

```
gnuplot> set ylabel "y"
```

- x 軸のメモリを変更する（1 から 5 まで，0.5 刻み）

```
gnuplot> set xtics 1, 0.5, 5
```

- グリッド（格子）を表示する

```
gnuplot> set grid
```

演習 1

- $0 \leq x \leq 2, y = 2xe^x + e^{-x} * \cos(x) + e^{-x}\sin(x)$ をプロットせよ .
- グラフにタイトル "Example" をつける .
- x 軸にラベル "x" をつける .
- y 軸にラベル "y" をつける .
- exm.png として保存する .

演習 2

- 好きな式を探しプロットし，タイトル・軸のラベルをつけ png 形式で保存せよ．

参考

<http://www.ice.gunma-ct.ac.jp/~mame/kougi/kisoron/gnuplot.html> (2017.5.11)

<http://www.gnuplot-cmd.com/in-out/output.html>
(2017.5.11)