第08回: 可視化*

陰山 聡

神戸大学 システム情報学研究科 計算科学専攻

2014.06.12

*計算科学演習 I (2014 年前期) 3 号館 演習室

計算科学演習 |

第 08 回:可視化

gnuplot 入門

アニメーション

アンケート

課題

2 次元可視化

計算科学演習 |

第 08 回:可視化

2014.06.12 2 / 59

事務連絡

スケジュールの変更

- 6/12 データの可視化(陰山)
- 6/19 MPIを用いた並列計算 I (横川)
- 6/26 MPI を用いた並列計算 II (谷口)
- 7/03 MPI を用いた並列計算 III (横川)
- 7/10 休講
- 7/17 実践編 I (陰山)
- 7/24 予備日 実践編 II (?)(陰山)

可視化ソフトウェア

様々な可視化アルゴリズムを実装した便利なソフトウェアが多数ある。

- 市販可視化ソフト
 - IDL, AVS/Express, Tecplot, ...
- 無料可視化ソフト
 - ParaView, Vislt, Amira, Vapor, ...
- 大規模ソフトの可視化機能を使う
 - Mathematica, MATLAB, ...

• 基本ツール

• VTK, Visualization Library, ...



この演習では、gnuplot[†]を利用する。



発音は "new plot"



gnuplot とは[‡]

gnuplot is a command-driven interactive function plotting program. It can be used to plot functions and data points in both two- and three-dimensional plots in many different formats. It is designed primarily for the visual display of scientific data. gnuplot is copyrighted, but freely distributable; you don't have to pay for it.

Gnuplot is neither written nor maintained by the FSF...

. . .

演習室の環境設定

- π -computer にインストールされている gnuplot を使う
- グラフは (Unixの) X-Window (X11) で描かれる[§]
- 端末で X11 のクライアントを立ち上げる
- デフォルトでは外部の X11 アプリケーションからの描画は拒否する 設定なので、それを変更する必要がある。



gnuplot



演習室での設定手順

各自の端末で:

- 1. 全てのプログラム \rightarrow Xming \rightarrow Xming (特になにも起きない)
- 2. Tera term を立ち上げる
 - 2.1 → 最初のダイアログで「キャンセル」
 - 2.2 → Tera term の「設定」
 - $2.3 \rightarrow$ 「SSH 転送」
 - 2.4 \rightarrow リモートの (X) アプリケーションを … にチェックが入っていなけ ればチェック
 - 2.5 → ファイル → 「新しい接続」→ ログイン

(参考) Mac からの設定

- 1. X11を立ち上げる[¶]。
- 2. ssh -X my_id@pi.ircpi.kobeu.ac.jp
- 3. (π -computer 上で) gnuplot と入力

確認

以下のコマンドプロンプトが出るはず。

gnuplot>

ここで gnuplot> plot sin(x)

と入れてみよう。以下のようなグラフが表示されれば成功。



gnuplot のヘルプと終了方法

ヘルプは gnuplot のプロンプトで help と打つ。

gnuplotの終了はプロンプトで quit と打つ。

gnuplot の単項演算子^{||}

単項演算子			
記号	例	説明	
-	-a	マイナス符号	
+	+a	プラス符号 (何もしない)	
~	~a	*1の補数 (ビット反転)	
!	!a	* 論理的否定	
!	a!	* 階乗	
\$	\$3	* 'using' 内での引数/列指定	

^{||}gnuplot 4.4 のマニュアルより引用

gnuplotの二項演算子**

二項演算子			
記号	例	説明	
**	a**b	累乗	
*	a*b	積	
1	a/b	商	
%	a%b	* 余り	
+	a+b	和	
-	a-b	差	
==	a==b	等しい	
!=	a!=b	等しくない	
<	a <b< th=""><th>より小さい</th></b<>	より小さい	
<=	a<=b	以下	
>	a>b	より大きい	
>=	a>=b	以上	

**gnuplot 4.4 のマニュアルより引用

計算科学演習 |

第 08 回:可視化

&	a&b	* ビット積 (AND)
^	a^b	* ビット排他的論理和 (XOR)
I	alb	* ビット和 (OR)
&&	a&&b	* 論理的 AND
11	a b	* 論理的 OR
=	a = b	代入
,	(a,b)	累次評価
•	A.B	文字列の連結
eq	A eq B	文字列が等しい
ne	A ne B	文字列が等しくない

gnuplot の組み込み関数^{††}

数学ライブラリ関数				
関数	引数	戻り値		
abs(x)	任意	x の絶対値, x ; 同じ型		
abs(x)	複素数	x の長さ, $\sqrt{\operatorname{real}(x)^2 + \operatorname{imag}(x)^2}$		
acos(x)	任意	$\cos^{-1}x (\mathcal{P}-\mathcal{P} \dashv \mathcal{P} \dashv \mathcal{V})$		
acosh(x)	任意	ラジアンでの $\cosh^{-1}x$ (逆双曲余弦)		
arg(x)	複素数	xの偏角		
asin(x)	任意	$\sin^{-1}x$ (アークサイン)		
asinh(x)	任意	ラジアンでの $\sinh^{-1}x$ (逆双曲正弦)		
$\operatorname{atan}(\mathbf{x})$	任意	$ an^{-1}x$ (アークタンジェント)		
atan2(y,x)	整数または実数	$ an^{-1}(y/x)$ (アークタンジェント)		
$\operatorname{atanh}(\mathbf{x})$	任意	ラジアンでの $ anh^{-1}x$ (逆双曲正接)		
EllipticK(k)	実数 k ∈ (-1:1)	K(k) 第 1 種完全楕円積分		
EllipticE(k)	実数 k ∈ [-1:1]	<i>E</i> (<i>k</i>) 第 2 種完全楕円積分		
EllipticPi(n,k)	実数 n<1, 実数 k ∈ (-1:1)	$\Pi(n,k)$ 第 3 種完全楕円積分		

^{††}gnuplot 4.4 のマニュアルより引用

besj0(x)	整数または実数	jo ベッセル関数 (0 次ベッセル関数)
besj1(x)	整数または実数	j_1 ベッセル関数 $(1 次ベッセル関数)$
besy0(x)	整数または実数	y_0 ベッセル関数 (0 次ノイマン関数)
besy1(x)	整数または実数	y1 ベッセル関数 (1 次ノイマン関数)
$\operatorname{ceil}(\mathbf{x})$	任意	[x], x (の実部) 以上の最小の整数
$\cos(\mathbf{x})$	任意	$x or D = \psi d \cdot v \cos x$
$\cosh(\mathbf{x})$	任意	$\cosh x, x$ のハイパボリックコサイン
erf(x)	任意	erf(real(x)), x の 実部の誤差関数
$\operatorname{erfc}(\mathbf{x})$	任意	erfc(real(x)), 1.0 - (x の実部の誤差関数)
exp(x)	任意	e^x, x の指数関数
floor(x)	任意	x , x (の実部) 以下の最大の整数
gamma(x)	任意	gamma(real(x)), x の実部のガンマ関数
ibeta(p,q,x)	任意	ibeta(real(p,q,x)), p,q,xの実部の不完全ベータ関数
inverf(x)	任意	x の実部の逆誤差関数
igamma(a, x)	任意	igamma(real(a,x)), a,x の実部の不完全ガンマ関数
imag(x)	複素数	x の虚数部分 (実数)
invnorm(x)	任意	x の実部の逆正規分布関数

int(x)	実数	x の整数部分 (0 に向かって丸め)
lambertw(x)	実数	Lambert W 関数
lgamma(x)	任意	lgamma(real(x)), x の実部のガンマ対数関数
$\log(\mathbf{x})$	任意	$\log_e x, x$ の自然対数 (底 e)
$\log 10(x)$	任意	$\log_{10} x, x$ の対数 (底 10)
norm(x)	任意	x の実部の正規分布 (ガウス分布) 関数
$rand(\mathbf{x})$	任意	rand(real(x)), 疑似乱数生成器
real(x)	任意	xの実部
sgn(x)	任意	x > 0 なら 1, x < 0 なら -1, x = 0 なら 0. x の虚部は無視
$\sin(\mathbf{x})$	任意	$\sin x, x o t \lambda $
$\sinh(\mathbf{x})$	任意	sinh x, x のハイパボリックサイン
sqrt(x)	任意	\sqrt{x}, x の平方根
$\tan(\mathbf{x})$	任意	$\tan x, x$ のタンジェント
tanh(x)	任意	anh x, x のハイパボリックタンジェント

【演習】

$$f(x) = x^x$$

のグラフを描け。

タイトル

plot x**x title "x\^x"

複数のグラフ

カンマで区切る

plot sin(x) title "sin(x)", sin(sin(x)) title "sin(sin(x))"



様々なパラメータ(set コマンド)

set title "y=x^x"
set xlabel "x (no units)"
set ylabel "y (no units)"
plot x**x

定義域 と 値域、 replot

set xrange [0:5]

replot

グリッド表示

set grid replot



計算科学演習 |

関数の定義

```
s2(x) = sin(sin(x))
s4(x) = s2(s2(x))
s10(x) = s4(s4(s2(x)))
plot s10(x)
```



データのファイルからの読み込み

gnuplot には、ファイルに書き込まれた離散データを読み込み、それをグラフにする機能がある。



まずは準備

cp -r /tmp/viz/data_sample_1d [your_directory]

コード解説: data1d_generator.f95

ポイント

- moduleの活用
- 単精度・倍精度浮動小数点数の指定方法

【演習】

サンプル1D データの作成

make data.1d

data.1d



gnuplot の入力ファイルフォーマット

- # 以降はコメント
- 1行に x, y 値のペア
- デフォルトでは第1列が plot の x 座標、第2列が y 座標(変更可能)



gnuplot を立ち上げて gnuplot> plot 'data.1d' w lp としてみよう。

w lp は with linespoints の略で、線 (line)と点 (point)を表示すること を意味する。(w linespoints と書いてもよい。)

出力例



gnuplot スクリプト

- gnuplot ではコマンドプロンプトに手で入力する内容をファイルから読 み込ませることが出来る。⇒ gnuplot script

- さきほどコピーしたディレクトリの中に graph1d.gp がある。

```
#
 A simple sample gnuplot script: graph1d.gp
#
#
#
# set size square
                               # same side lengths for x and y
# set size 0.65, 1
                               # other way to set regular rectan
set xlabel 'x'
                               # x-axis
set ylabel 'func'
                               # y-axis
set xrange[-10:10]
                               # x coords range
set yrange[-1:1]
                               # y coords range
set title 'graph 1d'
plot 'data.1d' w lp
pause -1
     計算科学演習 |
                            第 08 回:可視化
                                                     2014.06.12
                                                             35 / 59
```

【演習】

gnuplot がまだ立ち上がっていたら quit コマンドで終了し、改めて shell で gnuplot graph1d.gp と入力。

Makefileを用意してあるので、 make graph としてもよい。

アニメーション

クイズ

 $f(x) = \sin(\sin(\sin(\dots\sin(x)\dots)))$ はどんなグラフか?

次の関数 recsin

$$\operatorname{recsin}(x,n) := \begin{cases} \sin(x) & \text{for } n = 1\\ \operatorname{recsin}(x,n-1) & \text{for } n > 1 \end{cases}$$

のグラフは、nを増やしていくとどう変わっていくか?

【コード解説】

recsin(x,n)を計算するプログラム anim_data_generator.f95 ポイント

- 再帰関数
- 連番ファイルの作成

【演習】

- ソースコード: anim_data_generator.f95
- 実行: make anim.data

連番ファイルが50個生成されたことを確認。

gnuplot によるアニメーション

- アニメーション=静止画の連続。
- gnuplot のスクリプトを使えばアニメーションも簡単にできる。
- pause t とするとt 秒待ってから表示。
- pause t 小数の t がサポートされるかどうかはプラットフォーム依存。

アニメーション用スクリプトの例

```
set xlabel 'x'
                             # x-axis
set ylabel 'func'
                             # y-axis
set xrange[-10:10]
                             # x coords range
set yrange[-1:1]
                             # y coords range
set title 'anim 1d'
plot 'data.1d.000' w lp
pause 5
plot 'data.1d.001' w lp
pause 0.5
plot 'data.1d.002' w lp
pause 0.5
plot 'data.1d.003' w lp
.
٠
pause -1 # Enter キーが押されるまで待つ
```

アニメーション

アニメーションサンプル

先週紹介されたシェルピンスキーギャスケットのプログラム 現在とは別のディレクトリに以下のファイルをコピー cp -r /tmp/viz/sierpinski [your_directory2]

ソースコード解説 sierp2.f95

ポイント

- omp section による並列計算
- ギャスケット生成過程全体を関数化(引数:何回乱数を振るか)
- 連番ファイルの生成(整数から3桁の文字列)
- moduleの利用(privateとpublic)

実行

ソースコード: sierp2.f95 ジョブスクリプト: enshu.sh (先週のもの)

- コンパイル: frtpx -Kopenmp sierp2.f95
- ジョブ投入: pjsub enshu.sh

上の一連の操作は make data で実行してもよい。 結果:連番ファイル (data.000 ... data.099)

data.??? の中身を見てみよう。

アニメーション用スクリプト

- 100 フレームのアニメーション plot コマンドを 100 行
- 手で書くのは大変!

スクリプト作成プログラム

make plotme.gp

plotme.gp が生成される。中身を確認。

実行: make graph

【演習】

make data

データ確認(100個[0番から99番]のファイルができるまで待つ) make graph

アンケート

アンケート

https://questant.jp/q/11PAHZMK

アンケート



課題

課題

課題

sierp2.f95 では

- 確率 1/3 で $(x^{n+1}, y^{n+1}) = (x^n/2 + 1, y^n/2)$
- 確率 1/3 で $(x^{n+1}, y^{n+1}) = (x^n/2 1, y^n)$
- 確率 1/3 で $(x^{n+1}, y^{n+1}) = (x^n/2, y^n + 1)$

としている。これを

- ・ 確率 1/10 で $(x^{n+1},y^{n+1}) = (x^n/2 + 1,y^n/2)$
- 確率 1/10 で $(x^{n+1}, y^{n+1}) = (x^n/2 1, y^n)$
- 確率 8/10 で $(x^{n+1}, y^{n+1}) = (x^n/2, y^n + 1)$

と変更したプログラム sierp3.f95 を作成し、その結果を gnuplot で確認 せよ。

課題

【提出方法】

diff sierp2.f95 sierp3.f95 | mail kage

2 次元可視化

2次元可視化

これまでは1次元データの可視化であった。 gnuplot では2次元データの可視化もできる。 plot ではなく splot コマンドを使う。

2D データフォーマット

x00 y00 関数値 x01 y00 関数値 x02 y00 関数値

•

٠

٠

x09 y00 関数値 (空行) x00 y01 関数値 x01 y01 関数値

x09 y01 関数値 (空行)

x09 y09 関数値

データサンプル

55 35 0.1165598588705999 56 35 9.9624877672293416E-002 57 35 8.1734108631726782E-002 58 35 6.2857224006520482E-002 59 35 4.2963409431420671E-002 60 35 2.2021479795155254E-002 61 35 0.0000000000000000 0 36 0.0000000000000000 1 36 2.1867122785152873E-002 2 36 4.2655284590767971E-002 3 36 6.2396502500601705E-002 36 4 8.1122529495226178E-002

٠

•

例1



例 2



例 3

