第11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

2013.07.04

*計算科学演習 I(2013 年前期) 3 号館 演習室

計算科学演習 |

第 11 回:可視化

2013.07.04 1 / 68



gnuplot 入門

事務連絡

予備

計算科学演習 |

第 11 回:可視化

2013.07.04 2 / 68



準備

heat1.f90の改訂

準備

heat2.f90

/tmp/130704/heat2.f90 をコピーせよ。

diff heat1.f90 heat2.f90

進備

```
heat2.f90
    c m ---> NGRID
    c nmax --> NLOOP
    + n_{-}mid
    c ...) /4.0_DP --> ...) *0.25_DP
    c \dots + heaat * h * h \longrightarrow \dots + heat_sq
   heat1.f90
11
program heat2
```

heat2.f90 の並列化

準備

heat3.f90

/tmp/130704/heat3.f90 をコピーせよ。

```
1
  heat3.f90
    + MPI, 1–D decomposition
    usage (on pi-computer)
     1) mpifrtpx heat3.f90
     2) pjsub heat3.sh
I
11
11
   heat2 f90
11
   c m ---> NGRID
11
   c nmax ---> NLOOP
н
   + n mid
!!
   c h=1.0_DP/m --> h=1.0_DP/(NGRID+1)
11
     c ...) /4.0_DP --> ...) *0.25_DP
!!
     c \dots + heaat * h * h \longrightarrow \dots + heat_sq
11
111
    heat1 f90
!!
```

ジョブスクリプト heat3.sh

進備

/tmp/130704/heat3.sh をコピーせよ。

```
#!/bin/bash
#P.IM -N "heat3"
#PJM -L "rscgrp=small"
#PJM -L "node=4"
#PJM -L "elapse=02:00"
#PJM −j
drawLine()
ł
  echo "#"{1..50} | sed 's/[ 0-9]//g'
}
drawLine
mpiexec ./a.out
```

```
drawLine
```

可視化とは

可視化の一般論

- 情報可視化 Information visualization
- データ可視化 Data visualization, Scientific visualization

1次元可視化

xの関数 f(x) を直感的に理解するためには、グラフを書くのが一番である。例えば、

$$y = f(x)$$

に対して、fの値域が定義域に含まれるとき、

$$f^{\times 2} := f((f(x)), \quad f^{\times 3} := f(f(f(x))), \quad \cdots$$

等と定義する。 $f(x) = \sin x$ に対する $f^{\times 10}(x)$ はどのような関数であろうか?



クイズ

xを0以上の実数として、xのx乗、つまり

$$f(x) = x^x \quad (x \ge 0)$$

はどんな関数であろうか?

- 最大値 / 最小値をとる x は?
- x = 0の時の値 $f(0) = 0^0$ は何だろう?

【後で演習】

2次元データの可視化:等高線

×とyの関数、つまり2次元の関数f(x,y)の形を理解するには、等高線 を描くのがよい。地表面での大気の圧力pの分布p(x,y)の等高線は天気 図でおなじみである。



Figure: 気象庁のウェブページ: http://www.jma.go.jp/jp/g3/より

計算科学演習 |

第11回:可視化

2013.07.04 14 / 68

等高線の描画アルゴリズム

ここで等高線の描画アルゴリズムを紹介しよう。計算格子点上に定義さ れたデータから一本の等高線を描くには、下の図のように短い線分をつ なげていけばよい。



-つの線分は4つの計算格子点で定義された長方形領域(セル)の中で 直線を描く。例えば f(x,y)=1.0 の値の等高線を描く場合を考えよう。

あるセルの4つの頂点におけるfの値が全て1.0よりも大きいか、あるい は1.0未満であれば、f=1.0の等高線はこのセルを通らない。



セルを周囲の4つの辺、それぞれの両端の頂点でのfの値が1.0を「挟め ば」その辺を等高線が通る。辺上のどの位置を等高線が横切るかは、線 形補間をすればよい。下の図はちょうど中点を通る例である。



このように一つのセルに対して行う処理を次々と順番に、全てのセル対して行えば、等高線ができあがる。

このアルゴリズムは marching squares と呼ばれる。

等高線とは、2次平面上に分布するスカラー場 *f*(*x*, *y*)を曲線の分布図に 変換して可視化する手法と言える。



3次元データの可視化:等値面

等高線の「3次元版」を考えよう。

3次元空間中の関数 f(x, y, z) がある値(例えば 1.0)をとる点の集合は、 方程式 f(x, y, z) = 1.0で決まる曲面である。

これを等値面という。



等値面を描くためのアルゴリズムとして Marching Cubes[†]がある。



 † このアルゴリズムは米国特許(US4710876A, 1985)であったが、既に失効している。

 計算科学演習
 第 11 回:可視化
 2013.07.04
 20 / 68

データ可視化アルゴリズムの分類

- 空間次元(1次元、2次元、3次元、それ以上)
- データの種類(粒子系データ、場のデータ)
- データの種類(スカラー場、ベクトル場、それ以上のテンソル場)

粒子系データの可視化例



3次元場の可視化アルゴリズム

- 【よく使うもの】
 - スカラー場
 - 等値面
 - 断面(2次元面)上の等高線・カラープロット
 - ボリュームレンダリング
 - ベクトル場
 - グリフ
 - 変位
 - 流線・流跡線





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開発さ れている。

- 市販可視化ソフト
 - IDL, AVS/Express, Tecplot, ...
- 無料可視化ソフト
 - ParaView, Vislt, Amera, 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。

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[¶]
- 端末の(マイクロソフトの)Windowsシステムで、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
 - $2.1 \rightarrow$ 「キャンセル」
 - 2.2 →「設定」
 - 2.3 → 「SSH 転送」
 - 2.4 → 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
 - $2.5 \rightarrow$ 「設定」→「設定の保存」→「保存」
 - 2.6 →「新しい接続」

(参考) UNIX 系システムからの設定手順

- 1. X11 が使えるようにする(普通は何もする必要はない)
- 2. Mac ではバージョンによってはオプションインストールが必要^{||}。
- 3. ssh -X my_id@pi.ircpi.kobeu.ac.jp

gnuplot の立ち上げ

- 上記の手順で X11 アプリケーションの「貼り付け」を許可した上で、
- (*π*-computer上で) gnuplot と打つ。

確認

以下のコマンドプロンプトが出れば gnuplot の立ち上げ成功。 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< td=""><td>より小さい</td></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 11 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	$\cos^{-1}x (\mathcal{P}-\mathcal{P} \dashv \mathcal{P} \dashv \mathcal{V})$		
$a\cosh(x)$	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(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	$\sin x, x o \psi 1 \lambda$
$\sinh(\mathbf{x})$	任意	sinhx, 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
$tanh(\mathbf{x})$	任意	tanh x, x のハイパボリックタンジェント



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

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。

この機能を使って heat3.f90 で計算した正方形領域の中心点での温度が、 ヤコビ法によって次第に収束して行く様子を可視化してみよう。

【演習】1次元グラフ

- (1) heat3.f90 をコンパイルし、実行せよ。
- コンパイルは mpifrtpx heat3.f90
- ジョブ投入は pjsub heat3.sh
- (2) (標準)出力ファイル heat3.o....の中身を確認せよ
- エディタで開くよりも more / less / head / tail コマンドで見る方が 早い。

(3) gnuplot を立ち上げ、コマンドプロンプトに plot 'heat3.o....' w lp と 入れよ。

lp は 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の入力ファイル

- # はコメント開始
- 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)

【演習】ラベルの調整

(1) 上の図の中で文字とグラフが重なって見にくいので、文字を消す。 gnuplot のコマンドプロンプトで2行入れる:

unset key

replot

replot

(2) 縦軸の表示範囲を調整をする。

gnuplot のコマンドプロンプトで2行入れる:

set yrange [0:0.35]

(3) 図全体のタイトルと、×軸、y軸の説明を入れる。

set title "Temperatute at the Center"

set xlabel "iterations"

set ylabel "temperature"

replot

出力例



最終状態の温度分布

heat3.f90 で計算された最終的な平衡温度分布を gnuplot で見てみよう。 正方形を真ん中で横に切る y=0.5 の線上での温度の × 分布をグラフに する。

グリッド番号 i ではなく、x 座標の値を書き出す。i 番目の格子点の x 座

$$x_i = h \times (i - n_{\mathsf{mid}})$$

という関係にある。

data ディレクトリ

これからの演習でデータファイルを多数生成するので、データファイル 出力専用のディレクトリを用意しよう。

- 例えばいまソースコードの置いていあるディレクトリと同じ高さに mkdir ../data

などとして data ディレクトリを作成すること。別の場所、名前でも構わ ないが、以後のサンプルコードではそれに応じて適宜 data ディレクトリ の名前を変更すること。

データをファイルに書き出すサンプルコード

/tmp/130704/heat3_print_final_x_prof.f90 をコピーせよ。

diff heat3.f90 heat3_print_final_x_prof.f90

```
! heat3_print_final_x_prof.f90
! + open/close file 10
! + print out cross section 1D data.
! + integer jcut (cross section for output)
! + function this_process_has()
! usage (on pi-computer)
! 1) mkdir ../data (unless there is already.)
! 2) mpifrtpx heat3_print_final_x_prof.f90
! 3) pjsub heat3.sh (share the jobscript with heat3.f90.)
```

演習

(1) heat3_print_final_x_prof.f90 をコンパイルし、実行せよ。

- コンパイル mpifrtpx heat3_print_final_x_prof.f90

- ジョブ投入 pjsub heat3.sh (同じジョブスクリプトを使う)

(2) うまくいけば ディレクトリ ../data/ に temp.final_profile_x という ファイルができているはず。

(3) more / less / head / tail コマンドで確認。

(4) ../data ディレクトリに移動し、gnuplot を立ち上げる

(5) plot "temp.final_profile_x" w lp

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

```
- 以下の内容を src/heat3_print_final_x_prof.gp に保存せよ。
```

```
# heat3_print_final_x_prof.gp
#
# final temperature profile at y=0.5 as a function of x
#
set xrange [-0.5:0.5]
set yrange [0:0.5]
set xlabel "x"
set ylabel "temp at y=0.5"
plot "../data/temp.final_profile_x" w lp
pause -1
```

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 (src ディレクトリで)

gnuplot heat3_print_final_x_prof.gp と打て。

結果例



解析解

これまで (heat1.f90 やそれ以降のサンプルコードで) 解いてきた問題 は、境界条件 $T(x=\pm 1/2,y)=T(x,y=\pm 1/2)=0$ の下での

$$\frac{\partial^2 T}{\partial^2 x} + \frac{\partial^2 T}{\partial^2 y} + 4 = 0$$

の解である。

今描いている y = 0 での温度分布の解析(近似)解は以下で与えられる。 $T(x) = 0.589 - x^2 - 0.151 \{\cos(\pi x) + \cosh(\pi x)\} + 0.0038 \{\cos(2\pi x) + \cosh(2\pi x)\}$

― この近似解の求め方については以下参照:

http://www.research.kobe-u.ac.jp/csi-viz/members/kageyama/docs/ central_temp.pdf

計算科学演習 |

第 11 回:可視化

2013.07.04 62 / 68

レポート課題(演習)

heat3_print_final_x_prof.gp を改訂し、自分の計算結果と前ページの解析解の二つのグラフを同時に描く gnuplot スクリプトを作れ。

【提出方法】 π -computer 上の mail で kage 宛てに送信 (スクリプトに日本語を入れないこと。)

mail kage < your_script_file

【提出期限】7/11(木)18:00

アンケート

今回はこの場で入力してもらいます。その前に …

- 1. 自分の学籍番号の桁に現れる数字を足せ。その和を n とする。(例: 135X204X ならば n = 1 + 3 + 5 + 2 + 0 + 4 = 15)
- nⁿ を3で割った余りに2を足して、それをmとせよ。
 Unix では echo "15¹⁵%3 + 2" | bc で計算できる。
- 3. その m を使い、Emacs で Ctr-u m Esc-x hanoi と打て。
- 4. 「それ」が終わった人から、アンケートに答えてください。

事務連絡

事務連絡

事務連絡

来週

来週は講義はなし。

自習。

この演習室は解放します。TA に質問してください。



予備



アニメーションによって収束の様子を確認しよう。そのためのデータ (連番つきファイル群)を書き出すためのプログラム

/tmp/130704/heat3_print_x_prof_for_animation.f90

をコピーせよ。

このプログラムをコンパイル+実行せよ。

ジョブスクリプトは同じ。

うまくいけば data ディレクトリに連番ファイルが出力されるはず。確認 せよ。