

可視化（後編）*

陰山 聡

神戸大学 システム情報学研究科 計算科学専攻

2013.07.18

準備

アニメーション

2次元可視化

準備

準備

- 前回と同様に Xming と Tera term の設定をすること

アニメーション

アニメーションデータ

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/130718/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 heat3.sh

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。

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サンプル

```
#  
# gnuplot script generated by heat3_animation_x_prof_gp_gener  
#  
  
set xlabel "x"           # x-axis  
set ylabel "temperature" # y-axis  
set xrange [-0.5:0.5]   # x-coordinate  
set yrange [0.0:0.5]    # temperature min & max  
plot "../data/temp.j=middle.0000" w lp  
pause 5  
plot "../data/temp.j=middle.0001" w lp  
pause 1  
plot "../data/temp.j=middle.0002" w lp  
pause 1  
.  
.
```


【演習】1次元グラフ アニメーション

- /tmp/130718/heat3_print_x_prof_for_animation_plotscript_generator.f90
をコピーせよ
- 変数 NGRID と counter_end をチェックせよ。
- gfortran heat3_print_x_prof_for_animation_plotscript_generator.f90
- ./a.out > anyname.gp
- ファイル anyname.gp の中身を確認する
- gnuplot anyname.gp で実行

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。

これから gnuplot を利用して2次元データの可視化を行う。

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
.
.
.
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
```

diff heat3_print_x_prof_for_animation.f90 heat4_print_final_2d_prof.f90

```
! heat4_print_final_2d_prof.f90
!   + subroutine print__profile_2d
!   c module constants --> module common
!   + type ranks_t :: p
!   + type span_t :: jj
!   - myrank, nprocs, left, right (combined into "p")
!   - jstart, jend (combined into "jj")
!   + function adjust_jstart_and_jend
!   + function set_prof_2d
```

2D データ出力ルーチン (前半)

正方形上 (x,y 平面上) に分布する温度をすべて書き出す。

```

subroutine print__profile_2d(p,jj,f)
  type(ranks_t), intent(in) :: p
  type(span_t), intent(in) :: jj
  real(DP), dimension(0:NGRID+1, &
    jj%stt-1:jj%end+1), intent(in) :: f
  real(DP), dimension(0:NGRID+1,0:NGRID+1) &
    :: f_global ! 2d prof to be saved
  integer :: counter = 0 ! has
    save attrib.
  type(span_t) :: jj2 ! used
    for f_global
  character(len=4) :: serial_num ! put
    on file name
  character(len=*), parameter :: base = "../data/temp.2d."
  integer :: i, j

```

2D データ出力ルーチン (後半)

```
jj2 = adjust_jstart_and_jend(p, jj)
write(serial_num, '(i4.4)') counter
f_global(:, :) = set_prof_2d(jj, jj2, f)
if ( p%myrank==0 ) then
    open(10, file=base//serial_num)
    do j = 0 , NGRID+1
        do i = 0 , NGRID+1
            write(10,*) i, j, f_global(i, j)
        end do
        write(10,*) ' ' ! gnuplot requires a blank line here.
    end do
    close(10)
end if
counter = counter + 1
end subroutine print__profile_2d
```

出力データの確認

../data ディレクトリ中の連番つきファイル temp.2d.????の中身は以下のようになっているはず。確認せよ。

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
4	36	8.1122529495226178E-002
.		

gnuplot スクリプトを作る

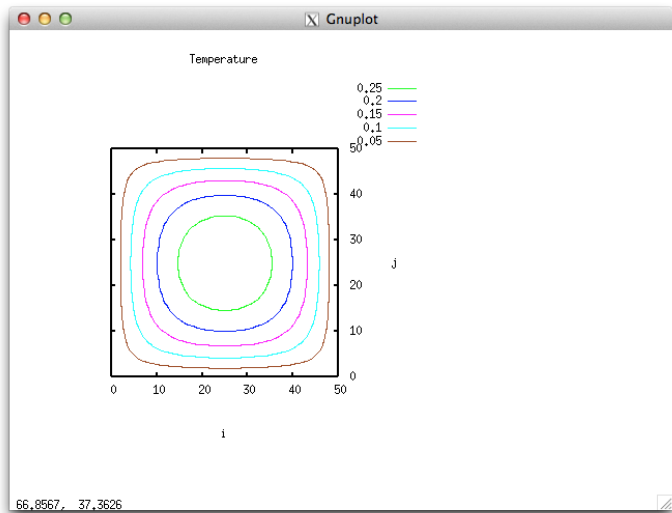
```
/tmp/130718/heat4_plot_contour_lines
```

```
#  
# A sample gnuplot script: heat4_plot_contour_lines.gp  
#  
# [ line contours ]  
#  
# set size square           # same side lengths for x and y  
set size 0.65, 1           # same side lengths for x and y  
set xlabel "i"             # x-axis  
set ylabel "j"             # y-axis  
set xrange[0:50]          # i-grid min & max  
set yrange[0:50]          # j-grid min & max  
set nosurface              # do not show surface plot  
unset ztics                # do not show z-tics  
set contour base           # enables contour lines  
set cntrparam levels 10    # draw 10 contours  
set view 0,0               # view from the due north  
set title "Temperature"  
plot "/data/temp.2d.0000" using 1:2:3 w l # with lines
```

【演習】2次元等高線の表示

- data/temp.2d.0000 のファイルに記された温度の分布を gnuplot の等高線で可視化してみよう。
- ファイル名 : heat4_plot_contour_lines.gp
- 実行方法 : gnuplot heat4_plot_contour_lines.gp
- ファイル名やパラメータ等を自由に変更してその効果を試せ。

結果の例



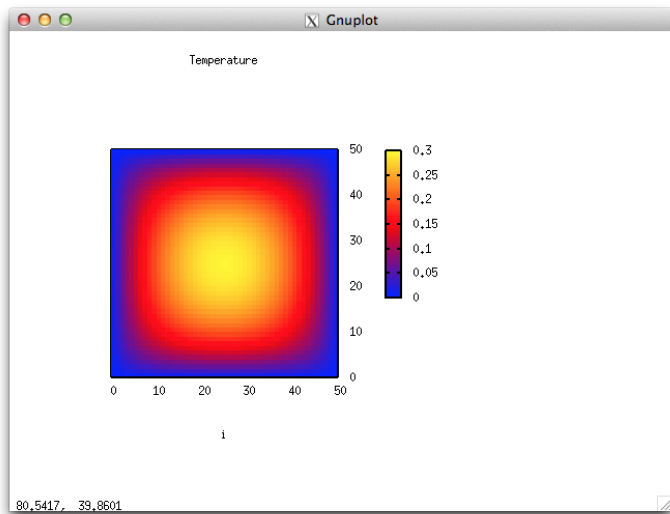
色分布による可視化（静止画）

- 等高線を描く代わりに正方形領域内部各点の温度を色で表現することも可能である。
- 実際に描いてみよう。
- gnuplot のサンプルスクリプトは次のとおり。

heat4_plot_contour_colors.gp

```
#  
# A sample gnuplot script: heat4_plot_contour_colors.gp  
#  
# [ color contours ]  
#  
# set size square          # same side lengths for x and y  
set size 0.65, 1          # same side lengths for x and y  
set xlabel "i"            # x-axis  
set ylabel "j"            # y-axis  
set xrange[0:50]          # i-grid min & max  
set yrange[0:50]          # j-grid min & max  
set palette defined (0 "blue", 0.15 "red", 0.3 "yellow")  
set nosurface              # do not show surface plot  
unset ztics                # do not show z-tics  
set pm3d at b              # draw with colored contour  
set view 0,0               # view from the due north  
set title "Temperature "  
plot "../data/temp.2d.0000" using 1:2:3  
pause -1
```

結果の例



gnuplot による鳥瞰図（静止画）

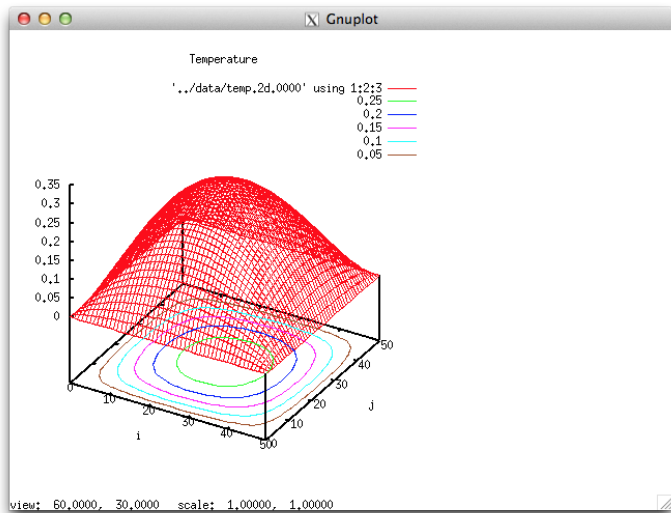
- 2次元温度分布 $T(x,y)$ を高さ (z) で表すことも可能である。
- この時の描画は3次的に行う必要がある。
- 空をとぶ鳥から見下ろしたような図は一般に鳥瞰図（bird's eye view）とも呼ばれる。

gnuplot スクリプト

ファイル名 : plot4_plot_birdseyeview.gp

```
#
# a sample gnuplot script: plot4_plot_birdseyeview.gp
#
# [ Bird"s Eye View ]
#
# set size square          # same side lengths for x and y
set size 0.65, 1
set xlabel "i"            # x-axis
set ylabel "j"           # y-axis
set xrange[0:50]         # i-grid min & max
set yrange[0:50]        # j-grid min & max
set contour base         # enables contour lines
set cntrparam levels 10  # draw 10 contours
# set palette defined (0 "blue", 0.15 "red", 0.3 "yellow")
# set pm3d                # draw with colored contour
set title "Temperature  "
splot "../data/temp.2d.0000" using 1:2:3 w l
pause -1
```


結果の例



gnuplot による鳥瞰図（回転のアニメーション）

gnuplot では view というパラメータで視線の方向（「鳥」がどの方向から見下ろしているか）を指定することが出来る。

この view パラメータを変更したアニメーションを作ってみよう。

gnuplot スクリプト生成プログラム

/tmp/130718/heat4_plot_rotating_birdseyeview_generator.f90