第1回オープンCAE講習会 OpenFOAM初中級講習B 2010年5月14日

離散化スキーム・線型ソルバー・ 境界条件の設定

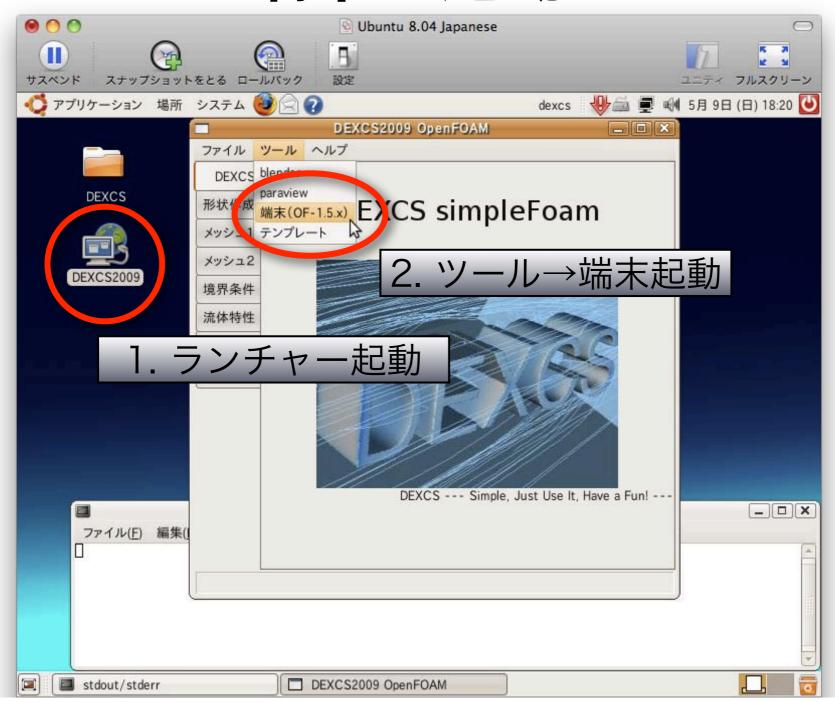
オープンCAE学会 今野雅(東京大学)

目次

- 1.下準備
- 2. 離散化スキームの設定変更
- 3. 線型ソルバーの設定変更
- 4. 境界条件の設定変更

下準備

端末の起動



端末の起動



チュートリアルの場所に行く

端末で赤字のように打ってみましょう!

```
run � cd tutorials/� ←ディレクトリ名はTab cd buoyantSimpleFoam/� キーで補完できます cd hotRoom/� ls �
```

出力

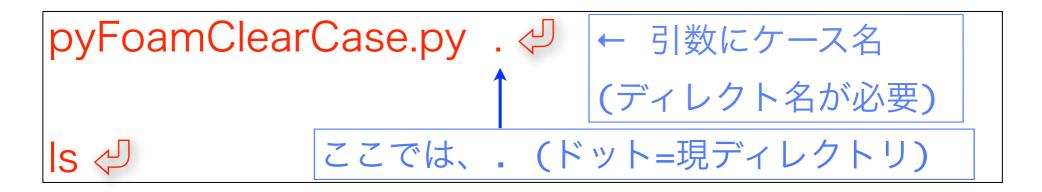
0	200	500 800	Allrun	log.buoya	antSimpleFoam	system
_		600 900	constant	log.se	etHotRoom	_
1000	400	700 Allclear	n log.bl	ockMesh	logs	

PyFoam

- ▶PyFoamとは?
 - •OpenFOAMの動作やデータを操作するPythonライブ ラリとユーティリティ群(43個)
 - •非標準だが、DEXCSではインストール済
- ▶主なユーティリティ
 - •pyFoamClearCase.py: 初期値以外の結果の削除
 - ・pyFoamPlotRunner.py: 方程式の残差や連続の式の 誤差をプロットしながら計算実行
 - ・pyFoamCloneCase.py: 計算結果以外のケースの複製

チュートリアル解析結果の消去

再計算するので解析結果を消します。



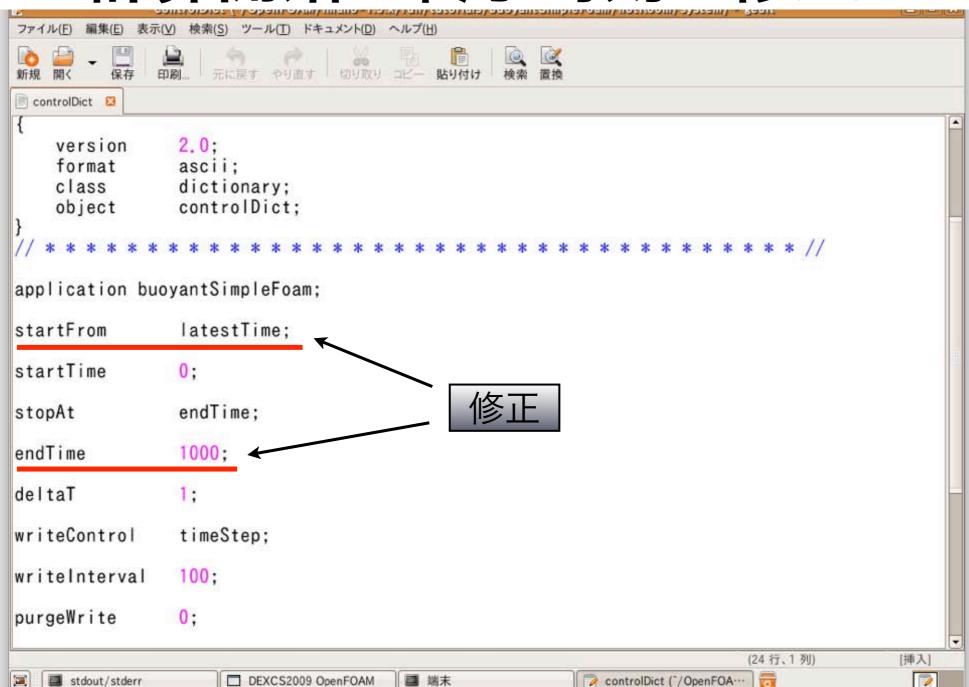
出力

- 0 Allclean Allrun constant logs system
 - ▶指定時刻以降の結果を消すことも可能
 - ▶並列計算の計算結果も消去可能

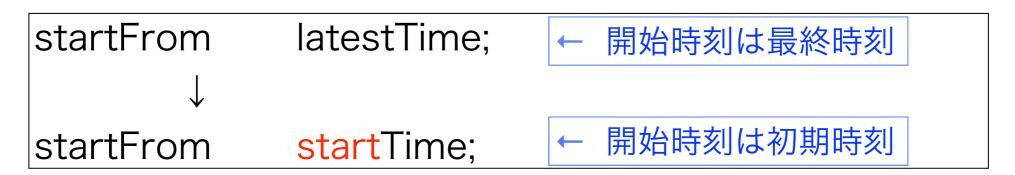
端末で赤字のように打ってみましょう!

←ファイル名はTabキー で補完できます

- ▶geditで起動して、ファイル名を選んでも良いですが、ファイル名を指定すると早いです。
- ▶もちろん、emacsやvi等、他の慣れているエ ディターを使っても構いません。



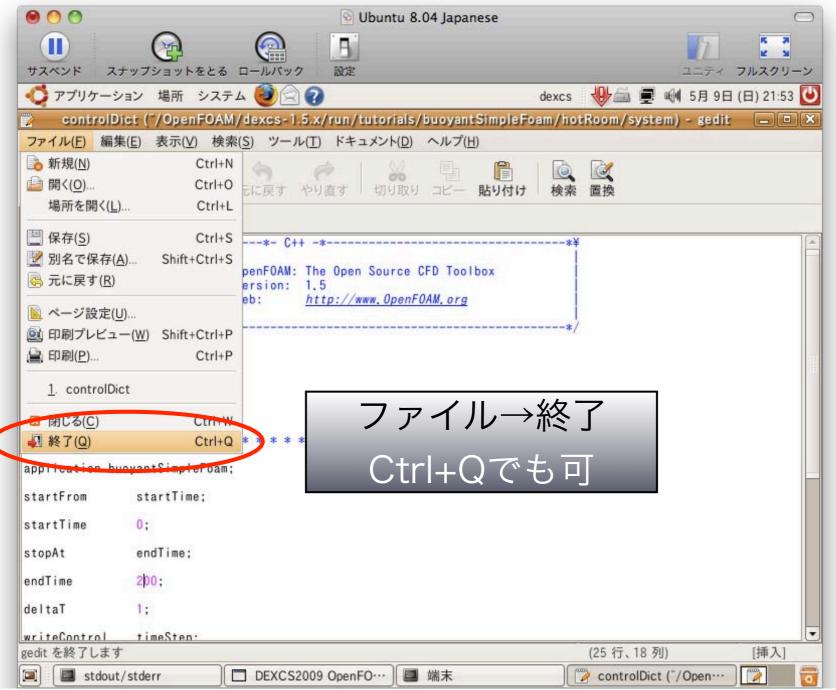
計算開始時刻を、初期時刻(=0)に固定

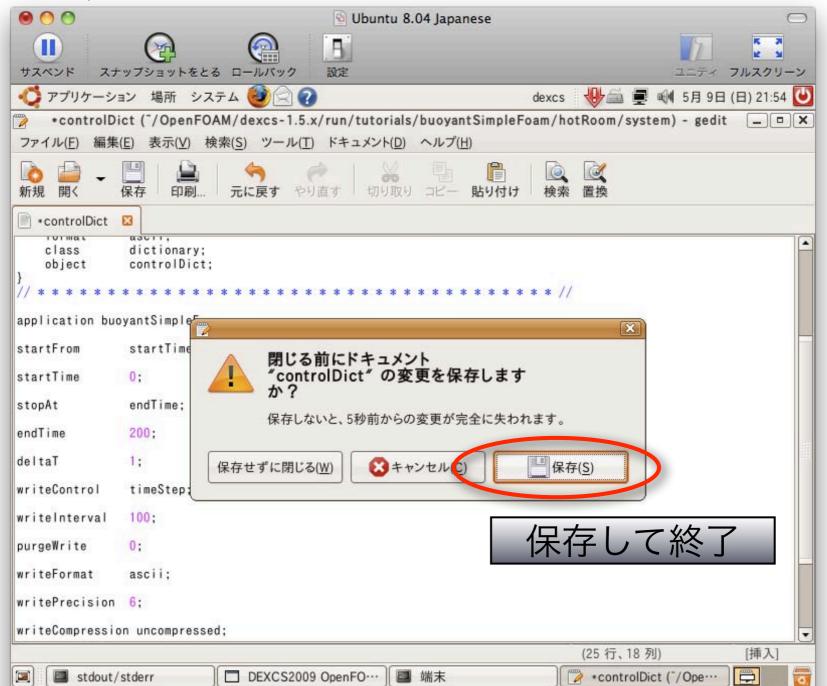


計算終了時刻を、200に変更

```
endTime 1000;

the properties of the properties
```





計算の実行と残差のプロット

端末で赤字のように打ってみましょう!

buoyantSimpleFoam 4

←コマンド名もTabキーで 補完できます

残差をプロットしながら実行

↑ ←カーソル↑(またはCtrl+P)で前のコマンド履歴が出ます

buoyantSimpleFoam Ctrl+A (trl+A(trl

←Ctrl+A(または**←**) で行頭に行きます

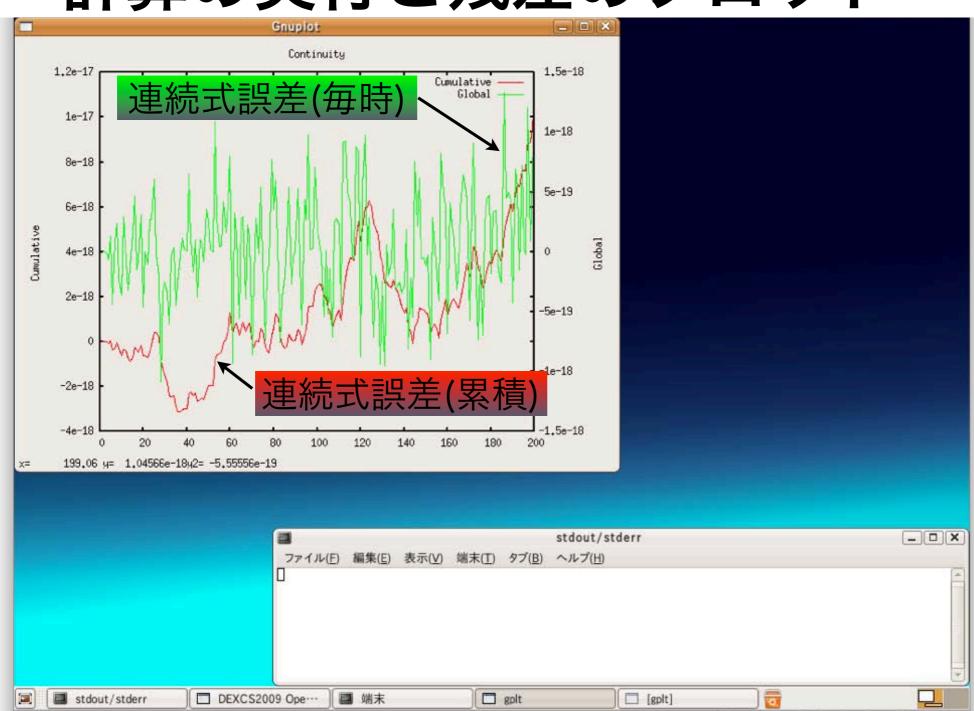
↓pyFoam...を打ちます

pyFoamPlotRunner.py buoyantSimpleFoam 4

計算の実行と残差のプロット



計算の実行と残差のプロット



結果の可視化

端末で赤字のように打ってみましょう!

paraFoam & 🕗

← コマンドの後に&を付けると、コマンドがバック・グラウンドで動き、続けて他のコマンドを実行できます。

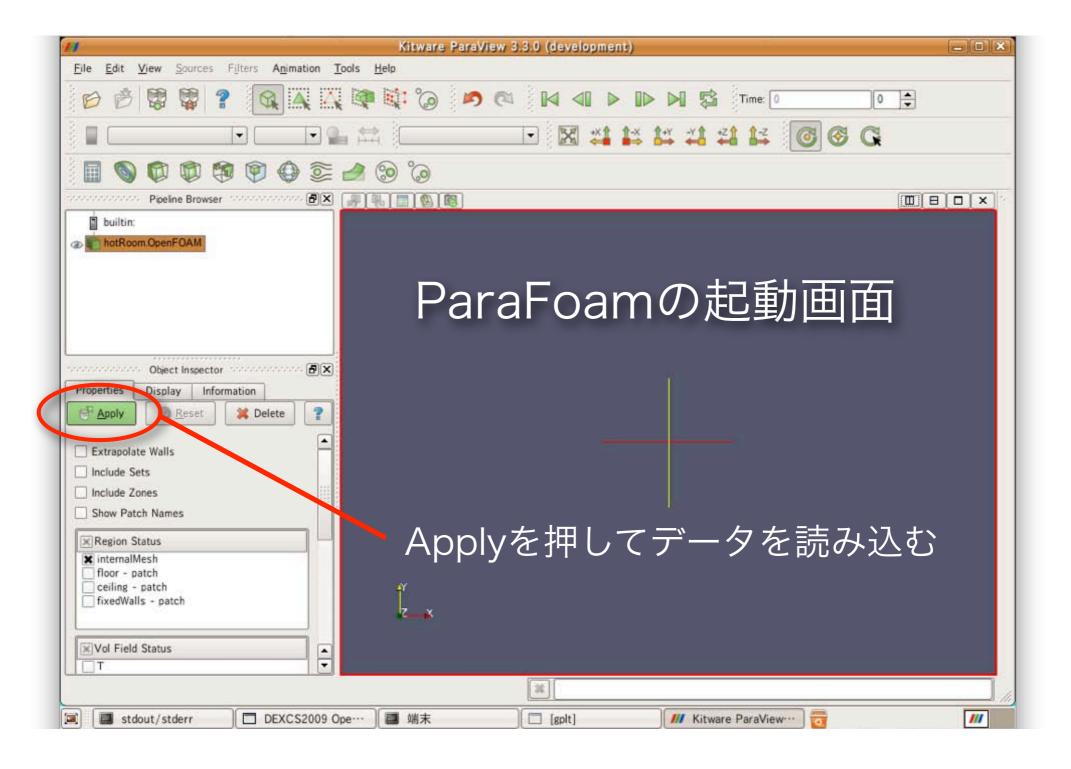
既にフォアグランドで実行していても...

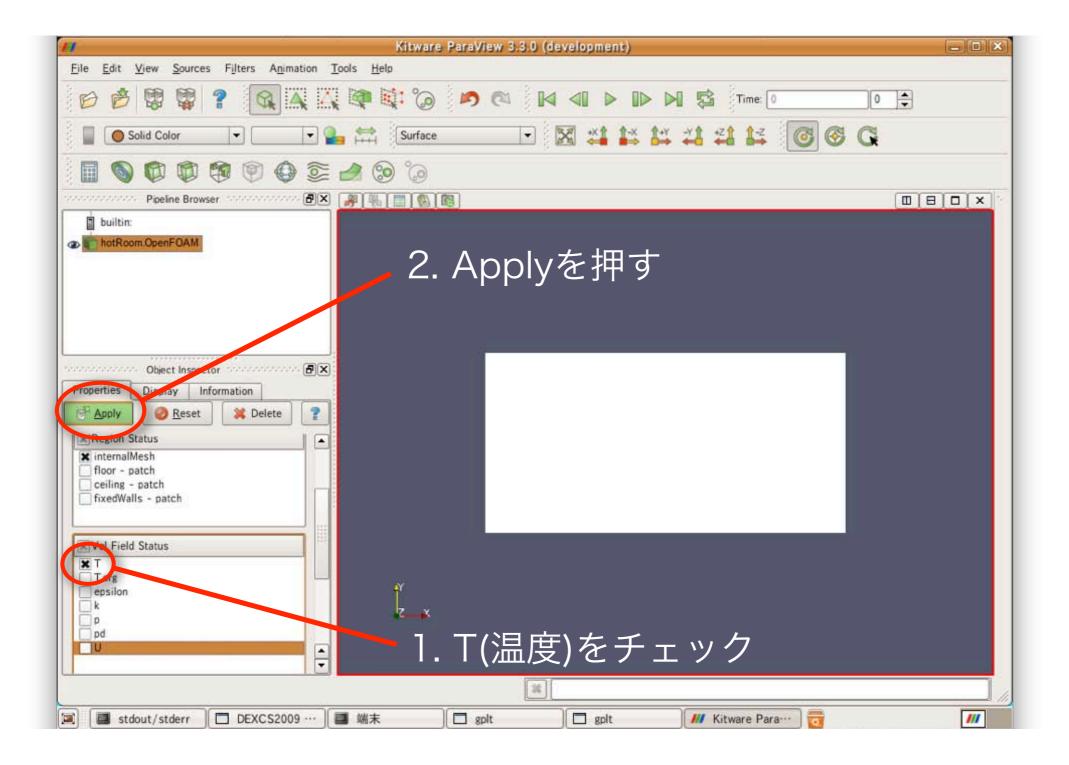
paraFoam 🕗

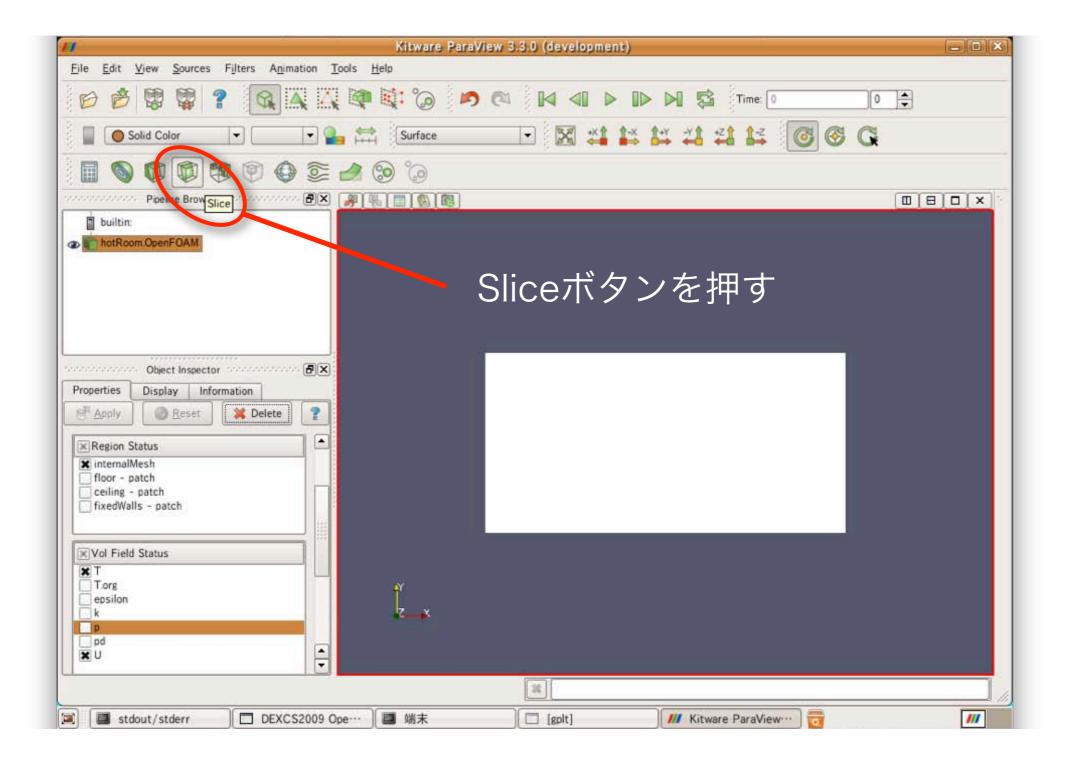
Ctrl+Z

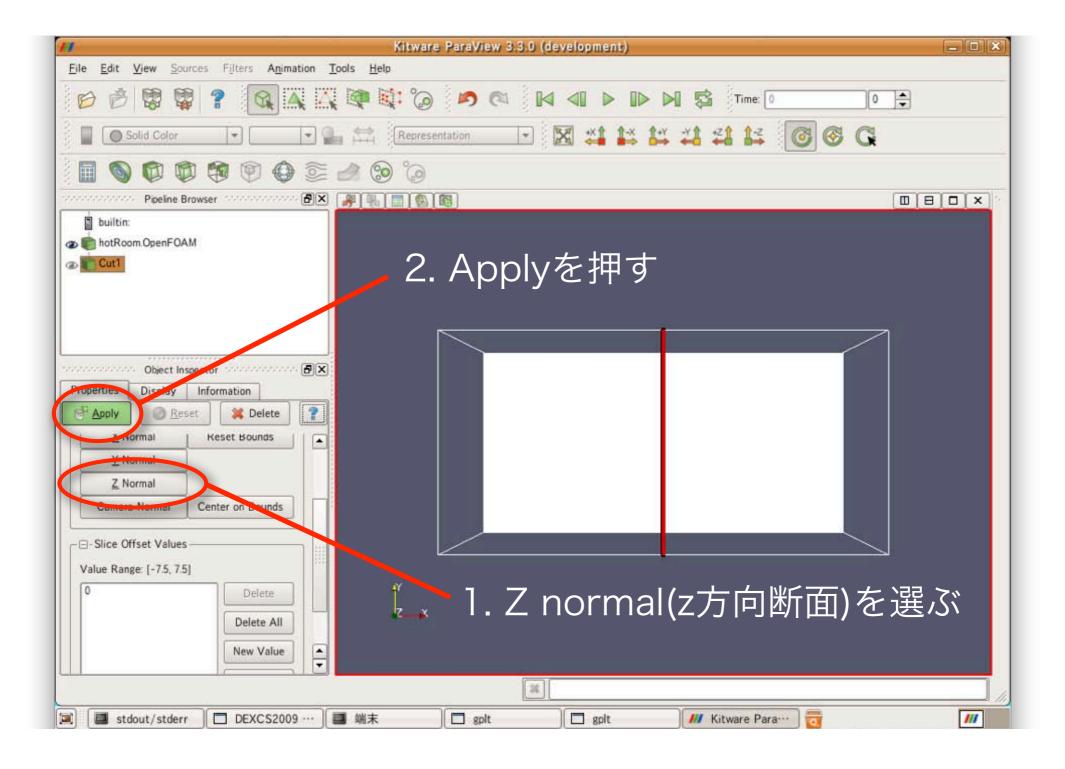
←Ctr1+Zで一度停止させてから、bgコマンドで バックグランド・ジョブにすることも可能です。

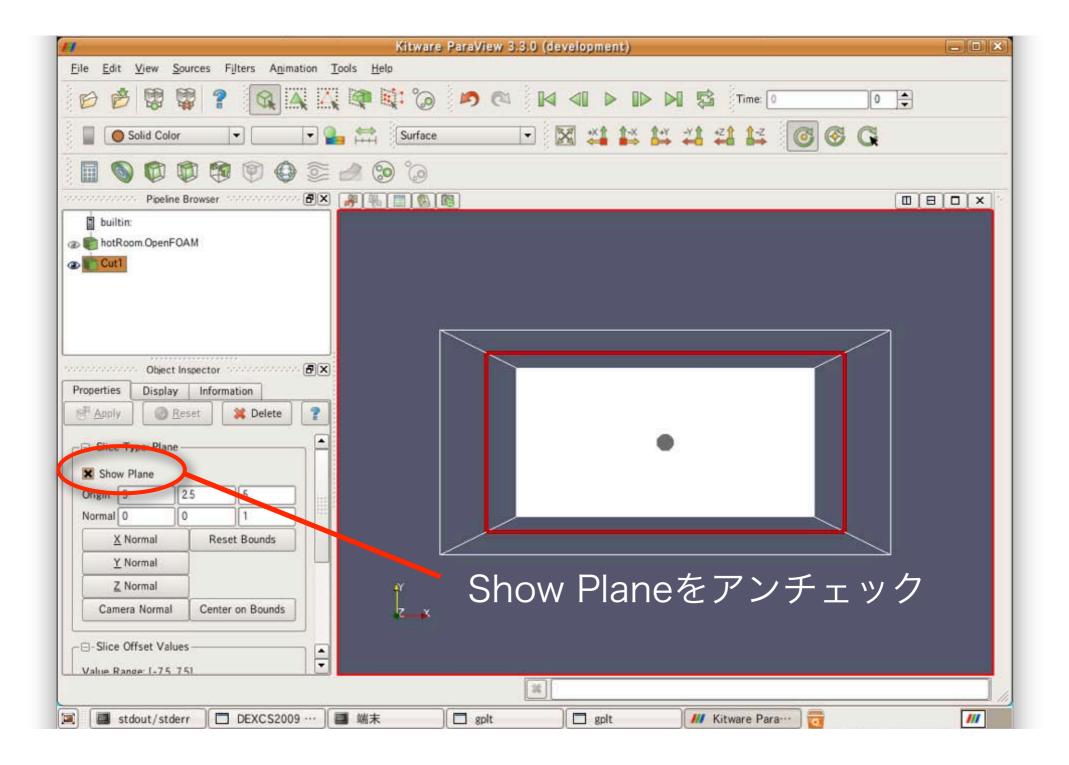
bg 却

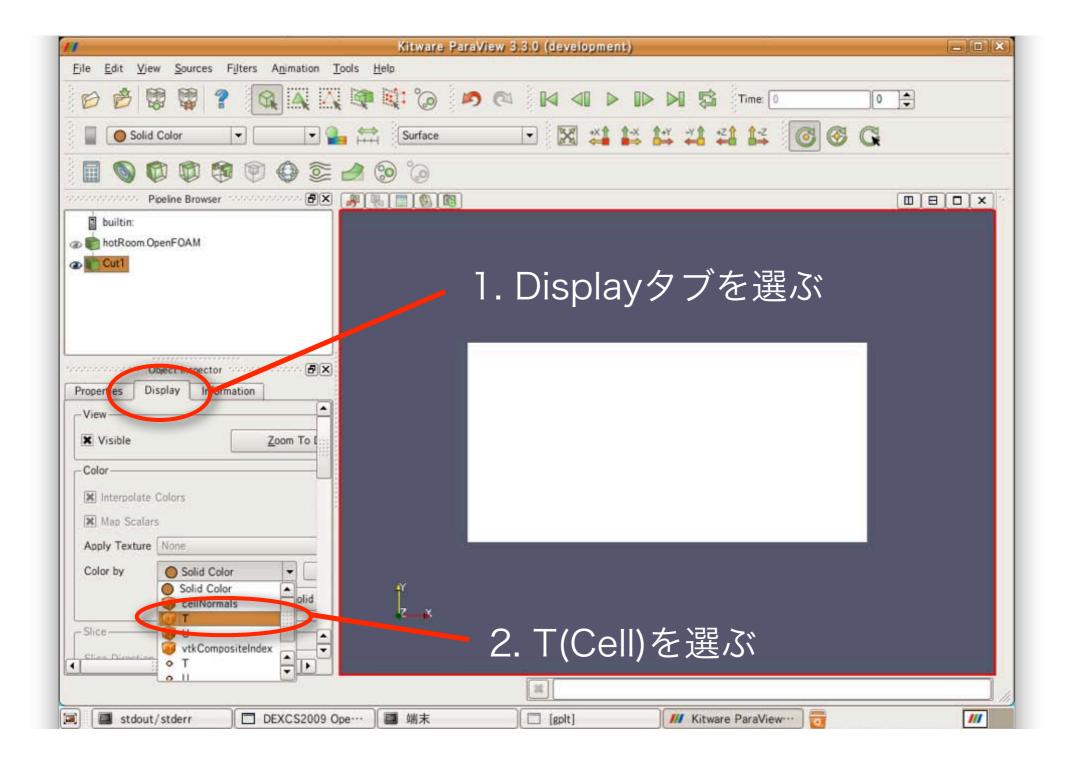


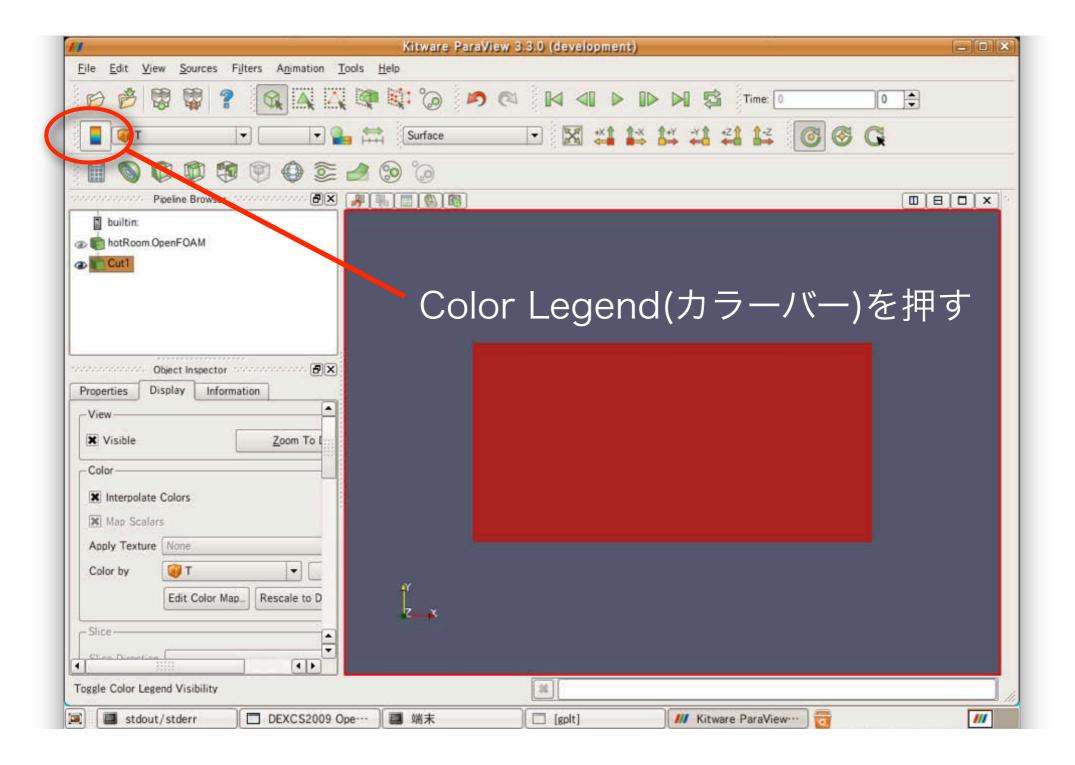


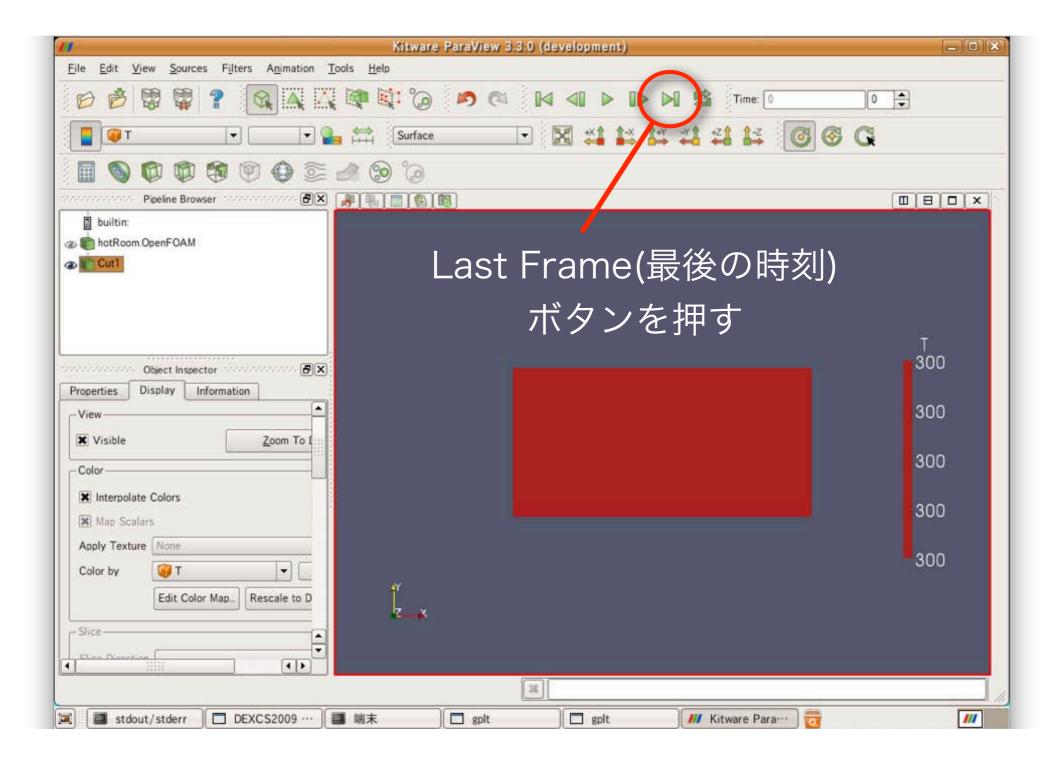


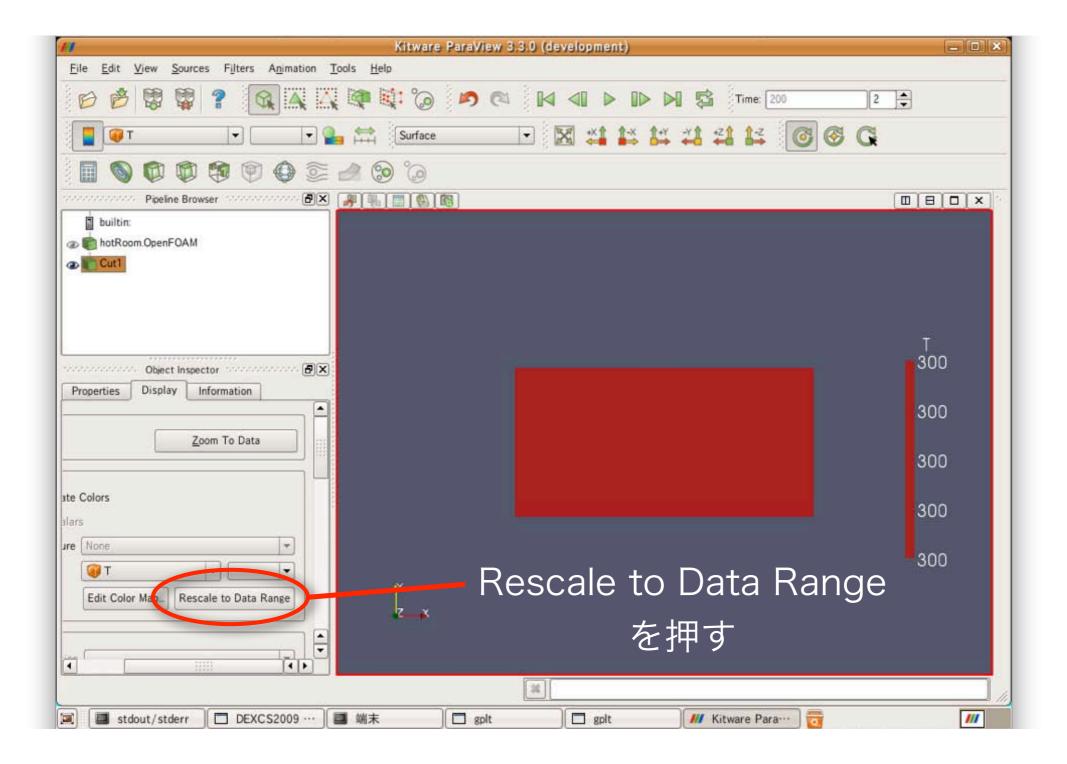


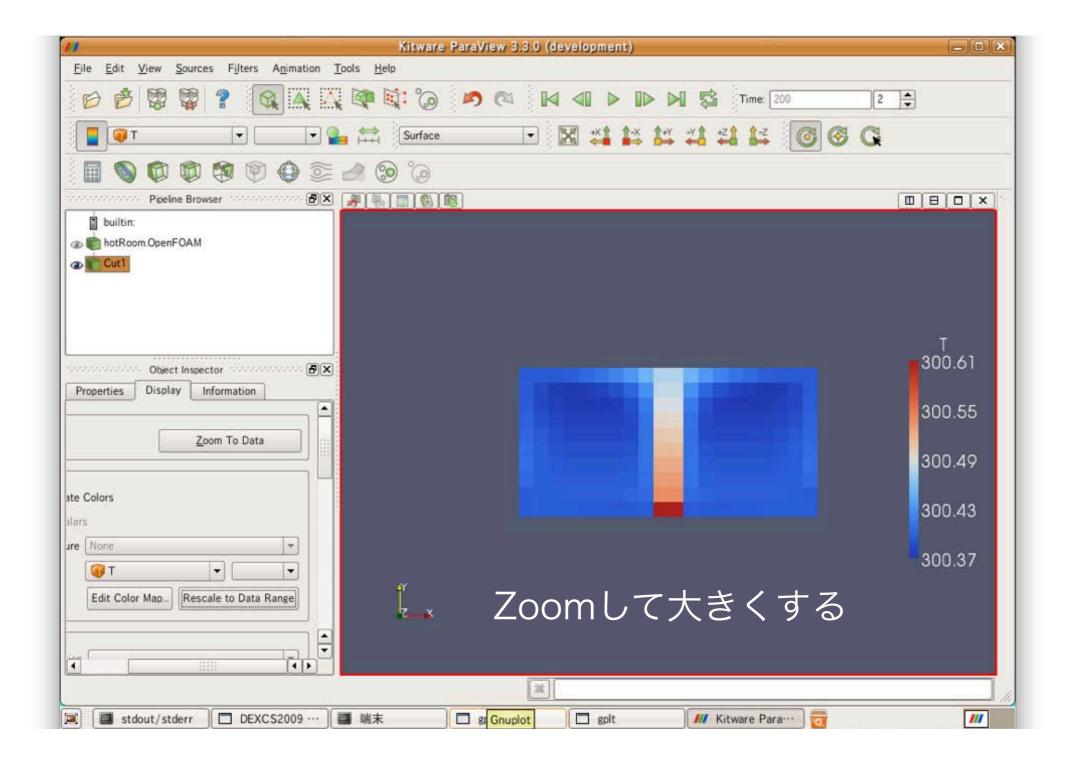


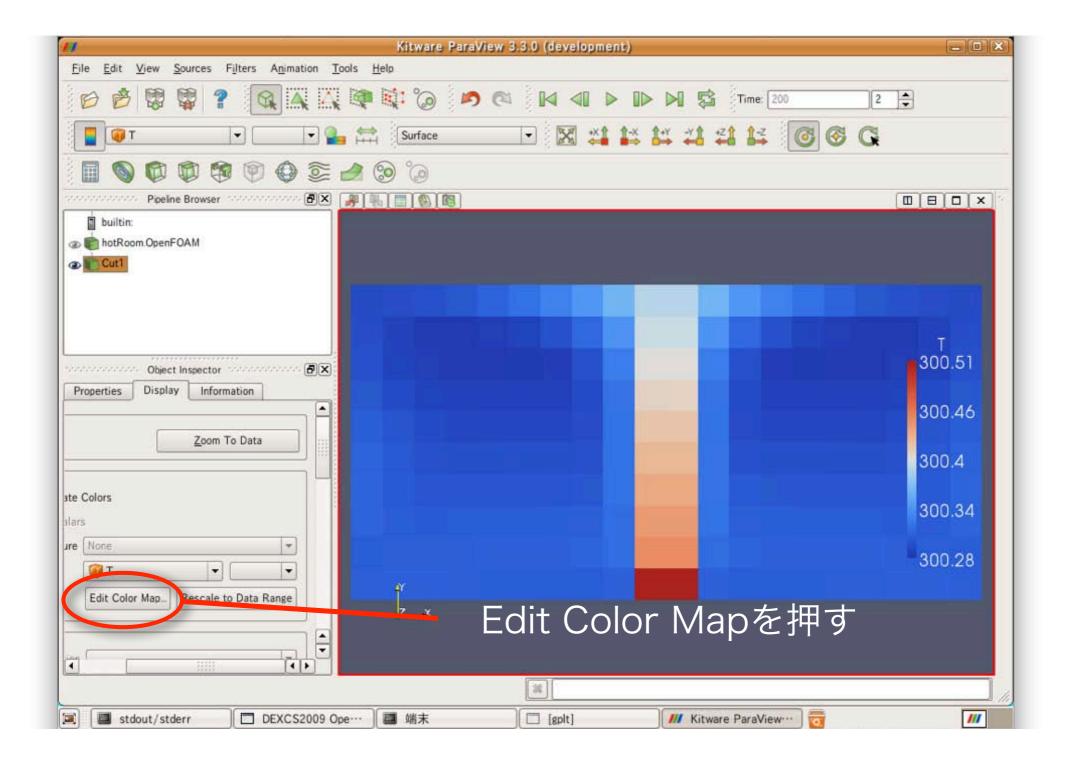


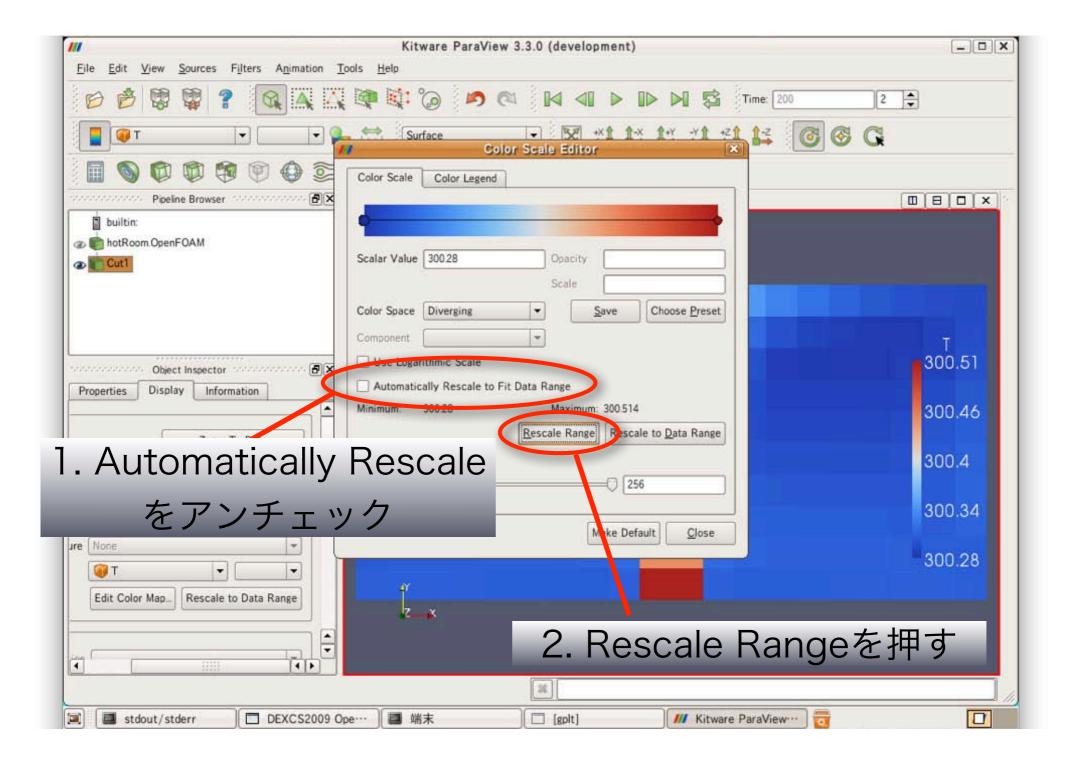


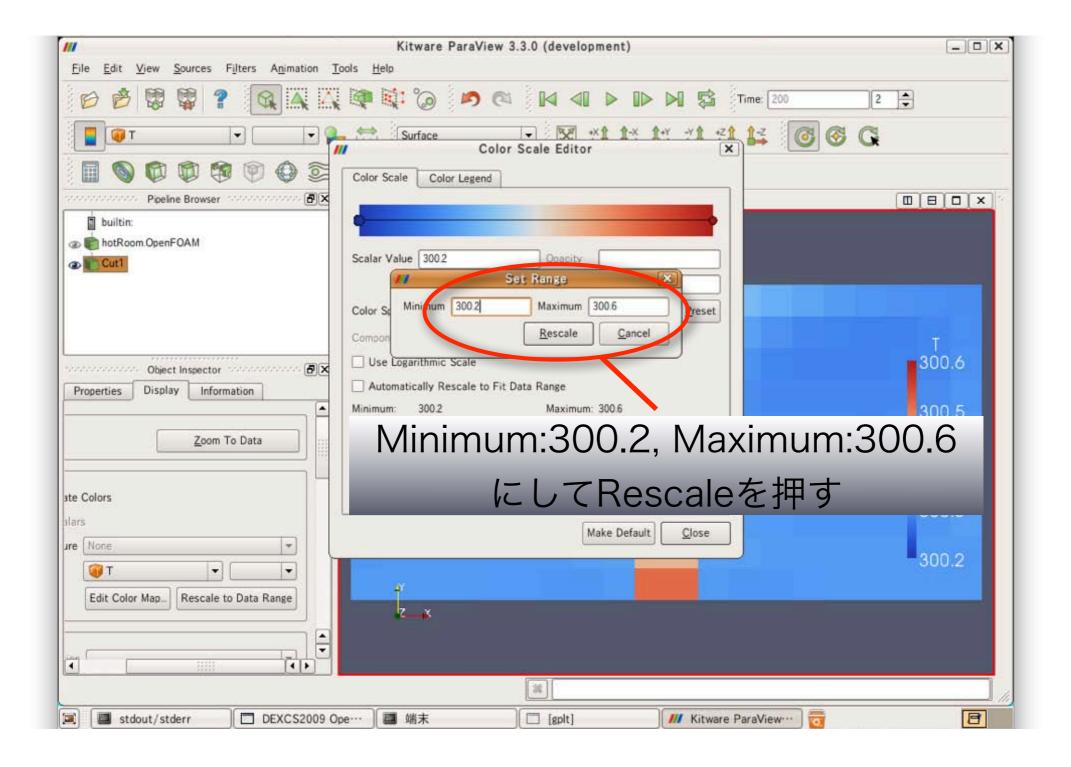


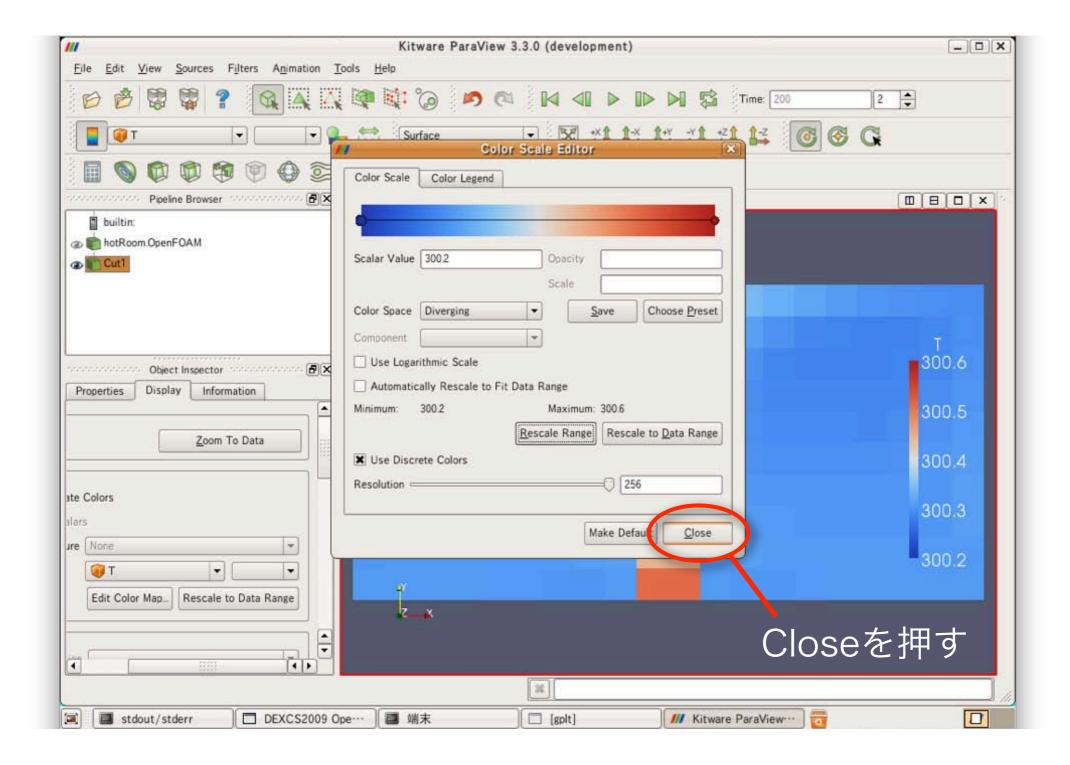


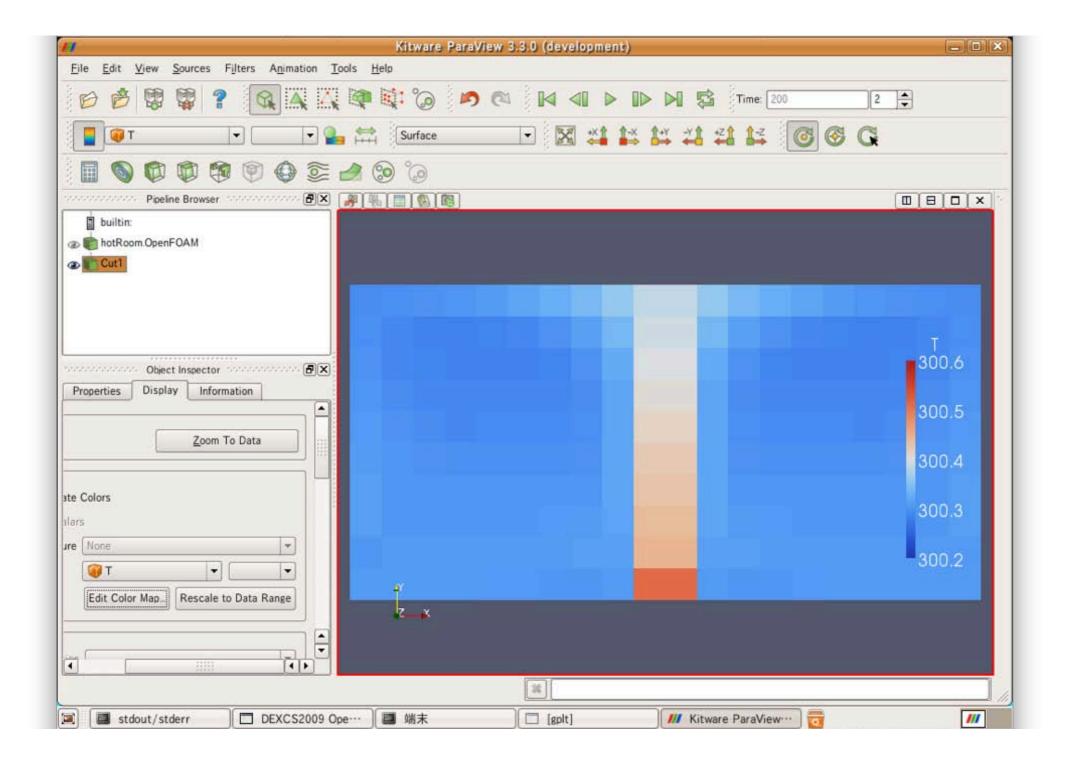












離散化スキームの設定変更

離散化スキームの変更

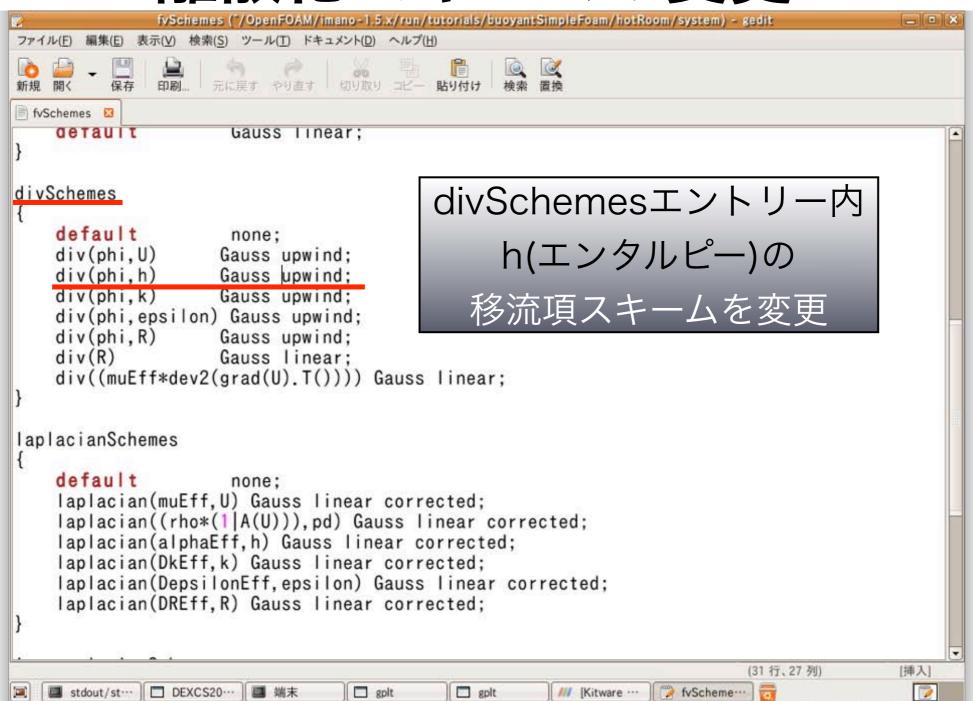
端末で赤字のように打ってみましょう!

gedit system/fvSchemes &

←ファイル名はTab キーで補完できます

スキームの変更と計算実行を繰り返し行うため、エディターを動かしたまま計算を実行できるように、最後に&を付けます。

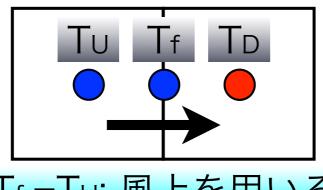
離散化スキームの変更



離散化スキームの変更

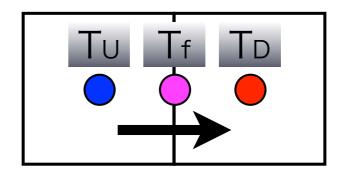
div(phi,h) Gauss upwind; ← 風上差分 (1次精度)
↓
div(phi,h) Gauss linear; ← 中心差分 (2次精度)

upwind(風上差分)



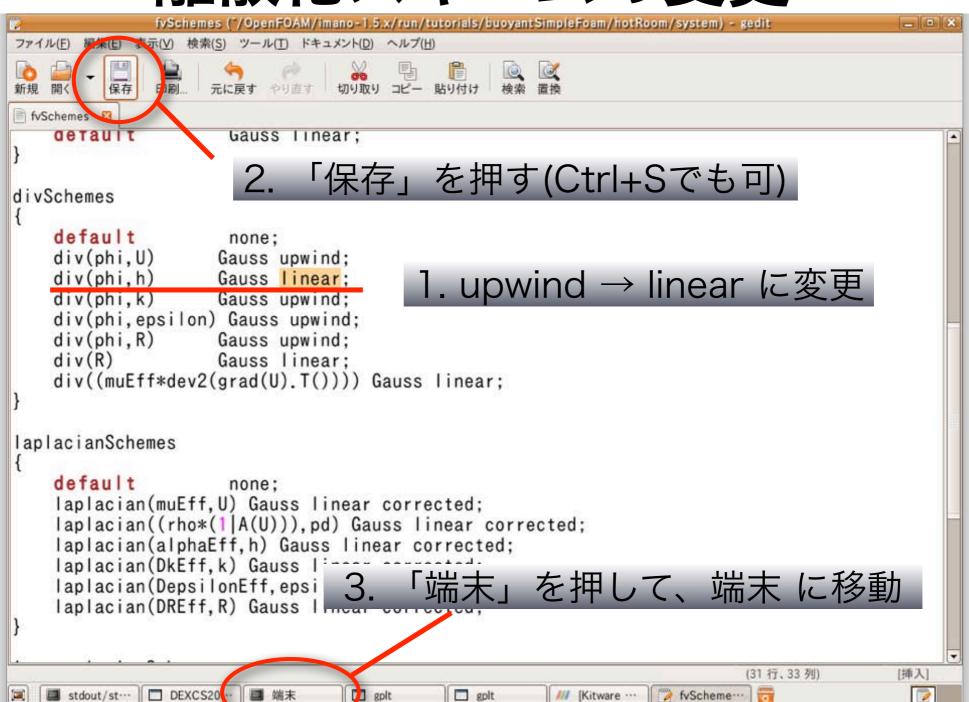
Tf =T∪: 風上を用いる

linear(中心差分)



Tf = (Tu +Tp)/2 :線型補間した中心の 値を用いる

離散化スキームの変更



離散化スキームの変更

端末で赤字のように打ってみましょう!

Ctrl+P

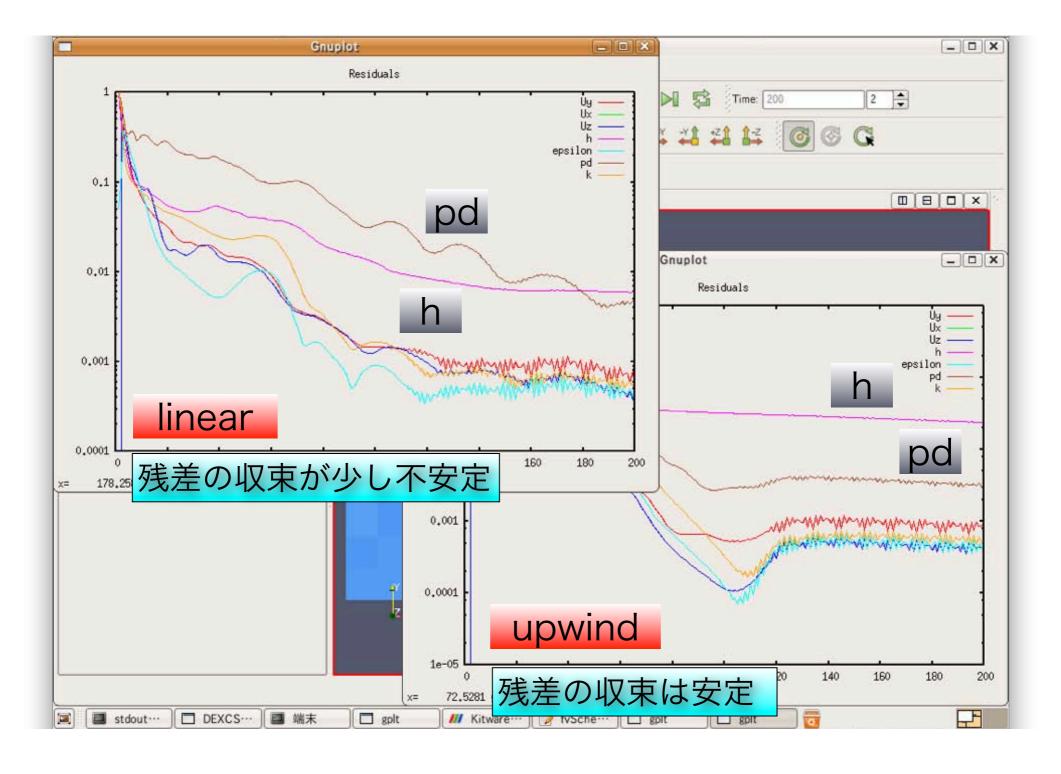
←Ctrl+P(または↑)で直前のコマンド履歴が出ます

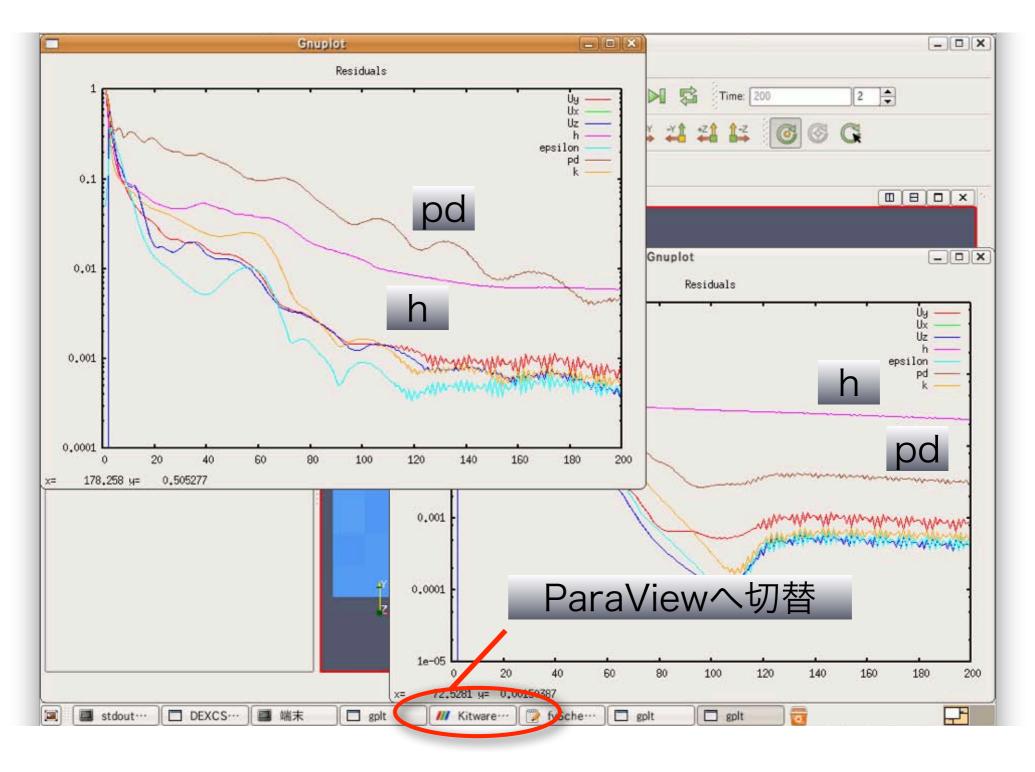
Ctrl+P

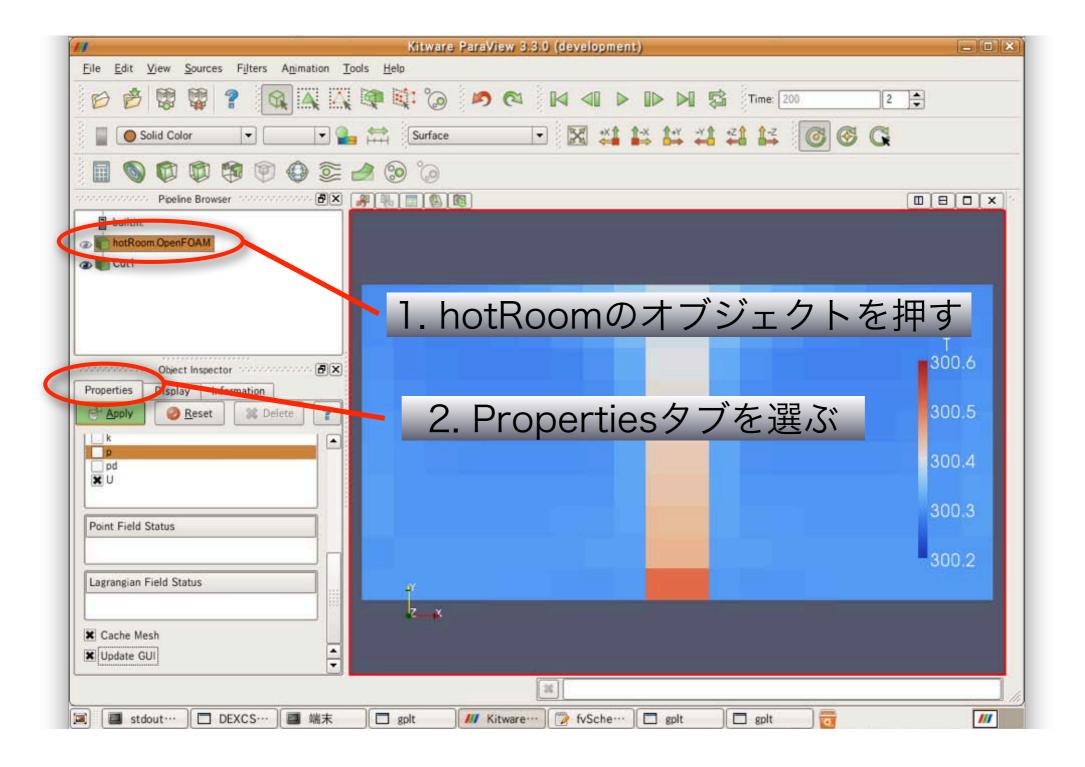
←何回かpyFoamPlot...の行が出るまで続けます

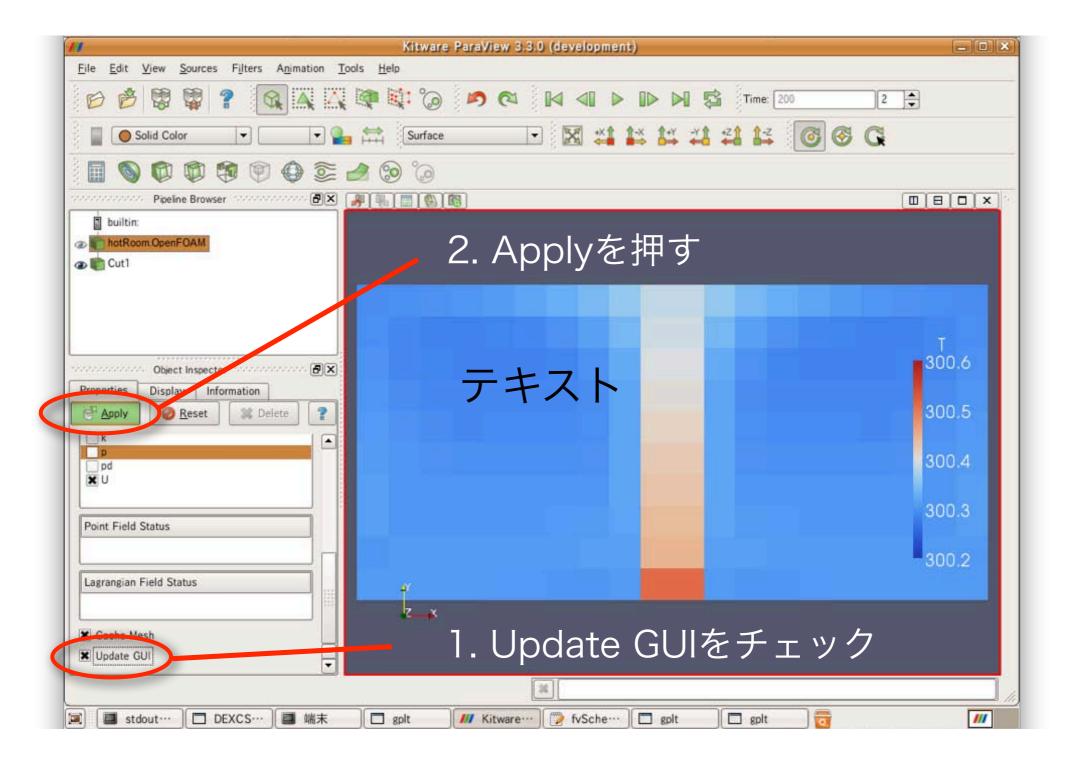
• • •

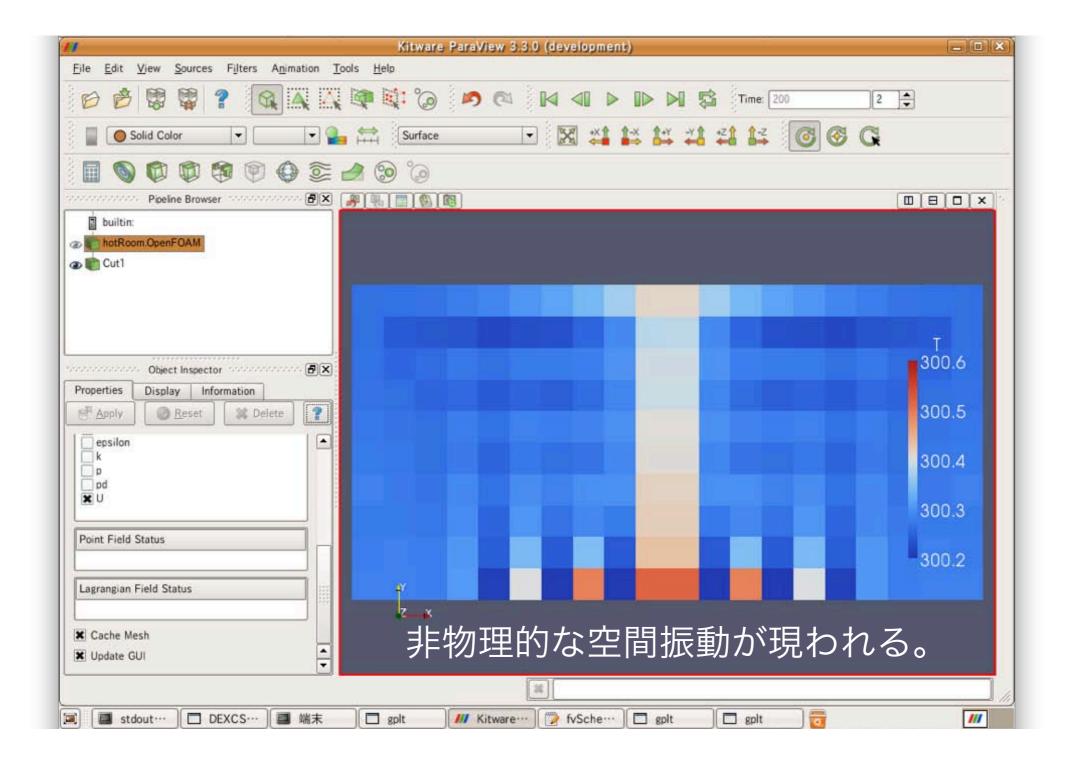
pyFoamPlotRunner.py buoyantSimpleFoam 🚚

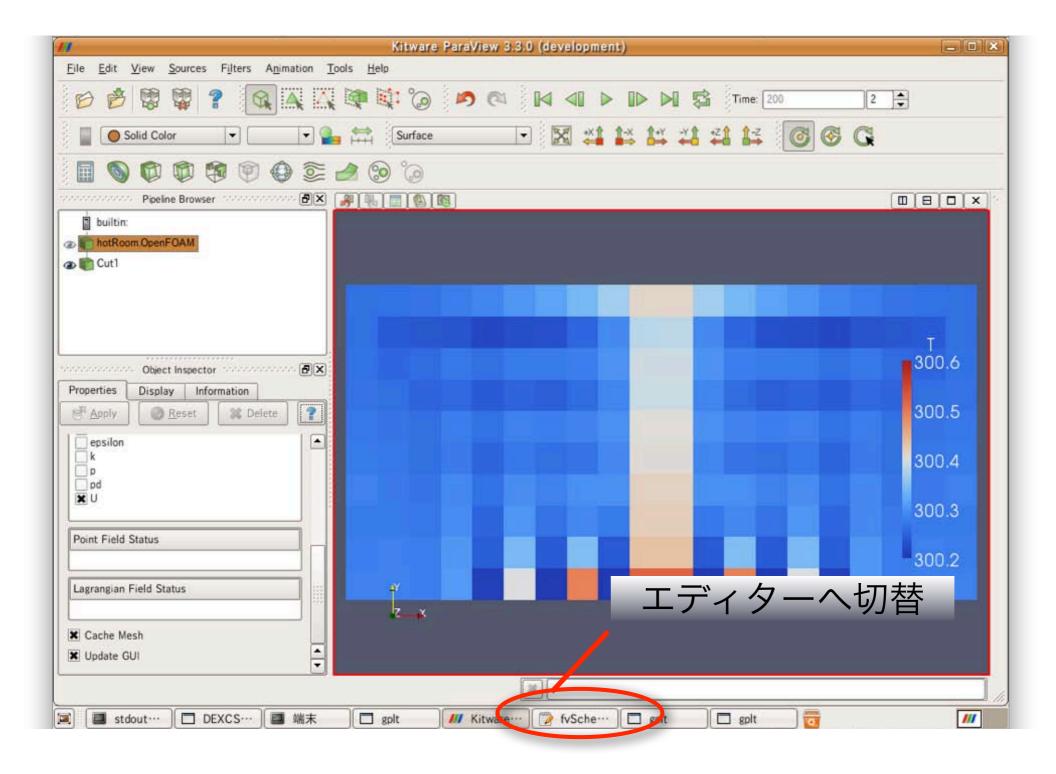




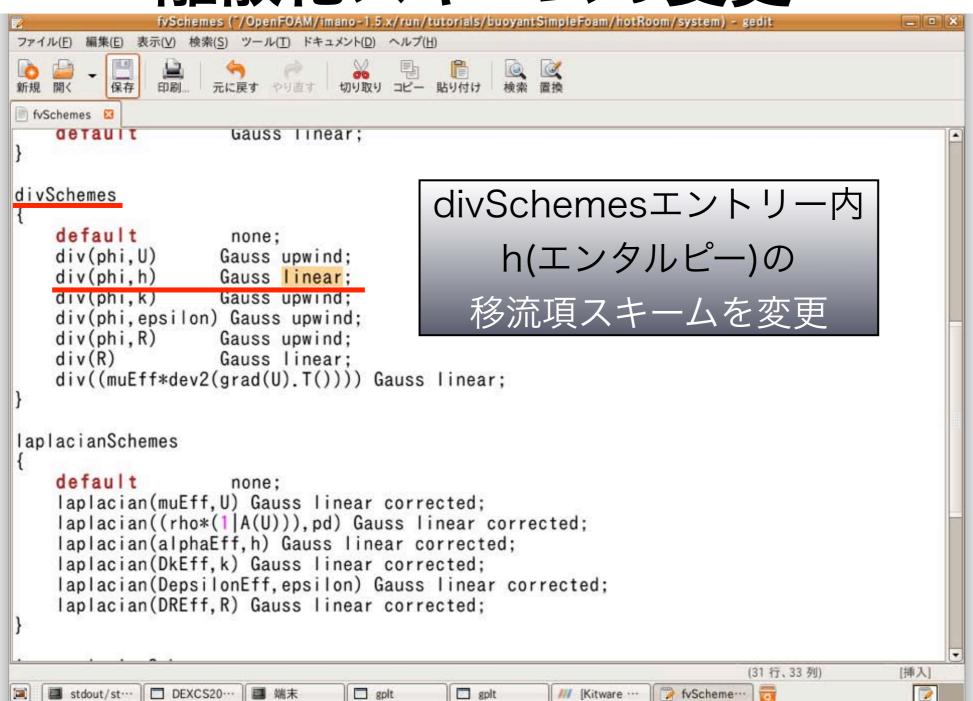








離散化スキームの変更



離散化スキームの修正

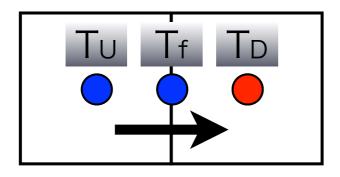
div(phi,h) Gauss linear;

← 中心差分 (2次精度)

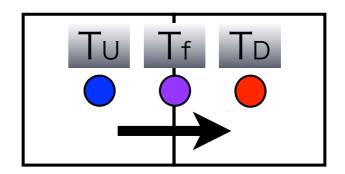
div(phi,h) Gaussauss limitedLinear 1; ←TVD法 (1~2次)

upwind(風上差分)

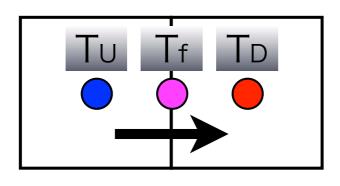




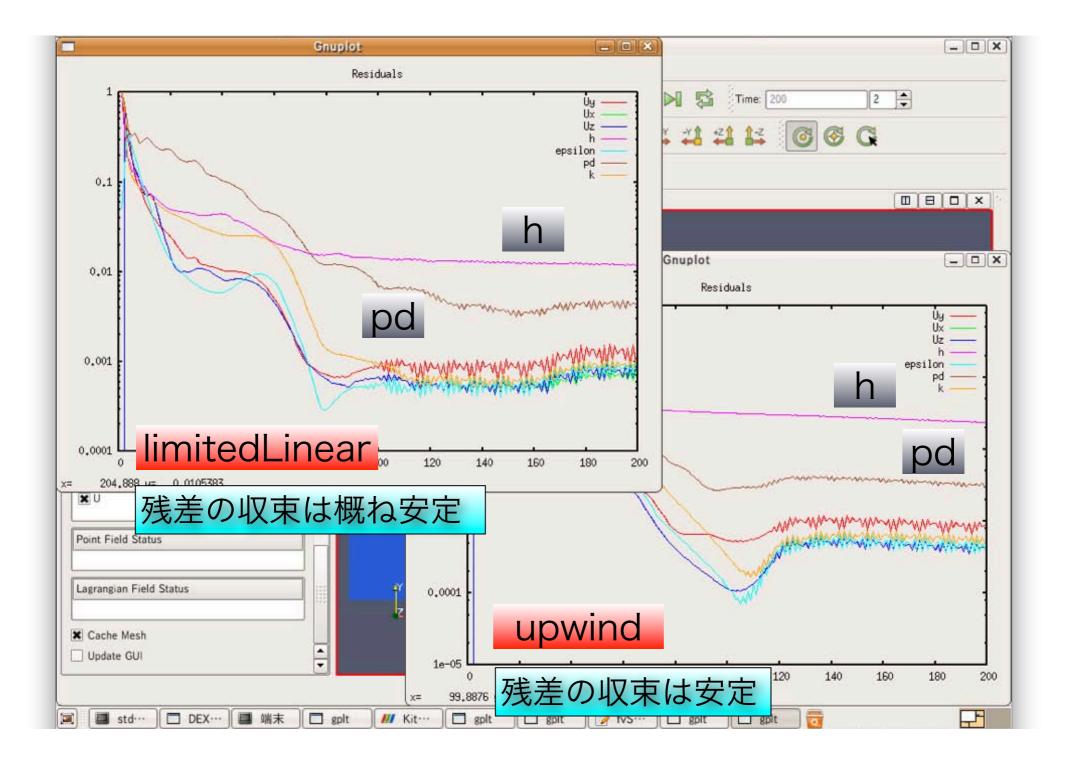
limitedLinear(TVD法)

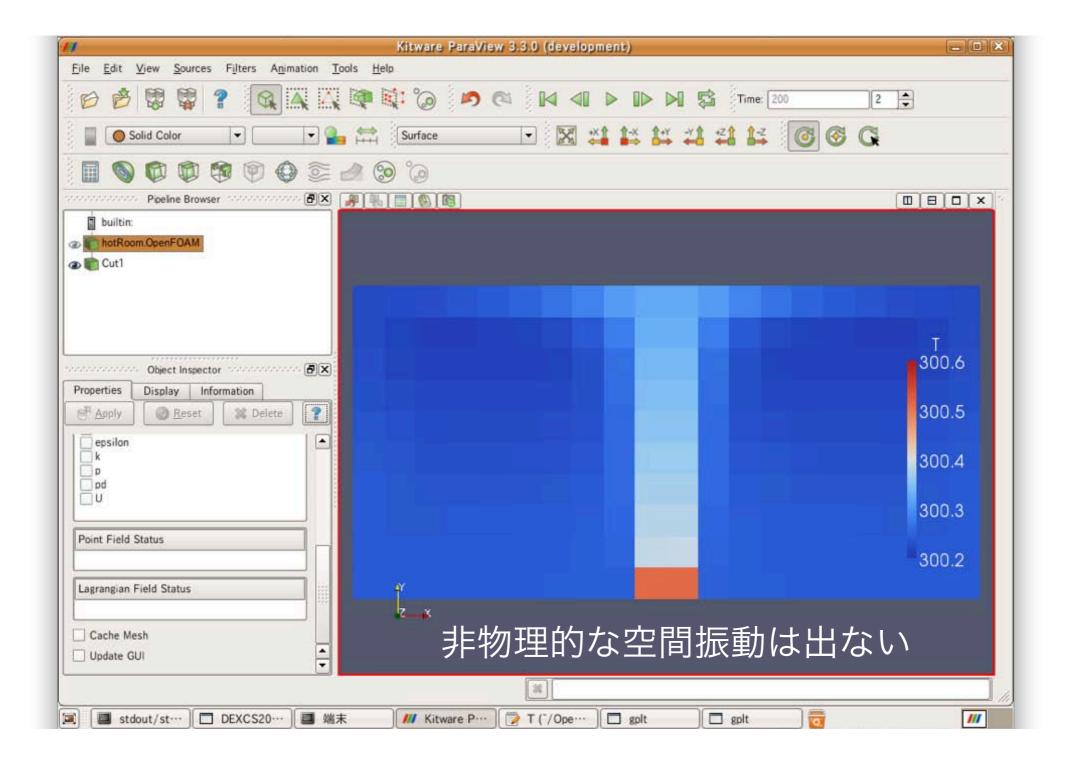


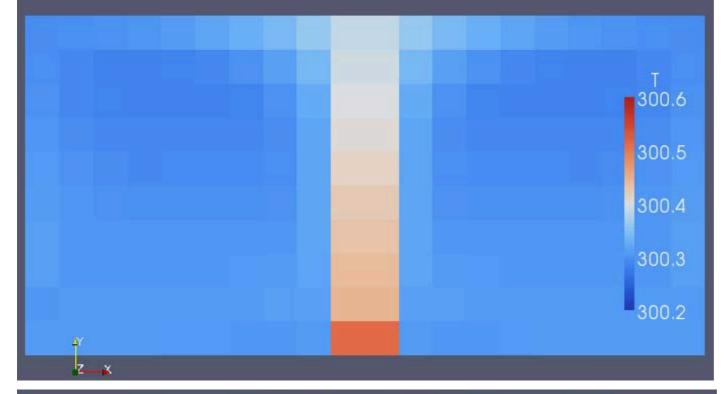
linear(中心差分)



 $T_f = \alpha T_U + (1-\alpha) T_D$ 安定性と精度を保つよう風上の値をブレンド

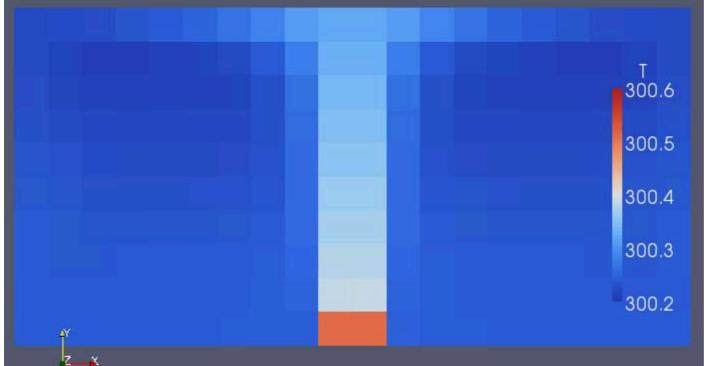






upwind

解の分布が拡散的

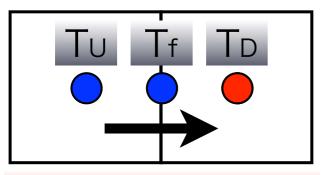


limitedLinear

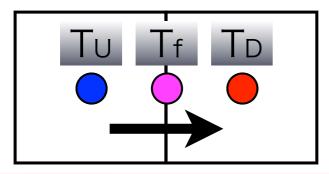
分布がシャープ

離散化スキームの比較

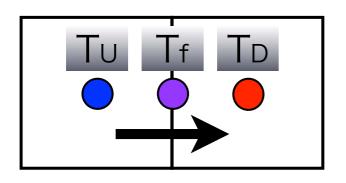
upwind(風上差分)



linear(中心差分)



limitedLinear(TVD法)



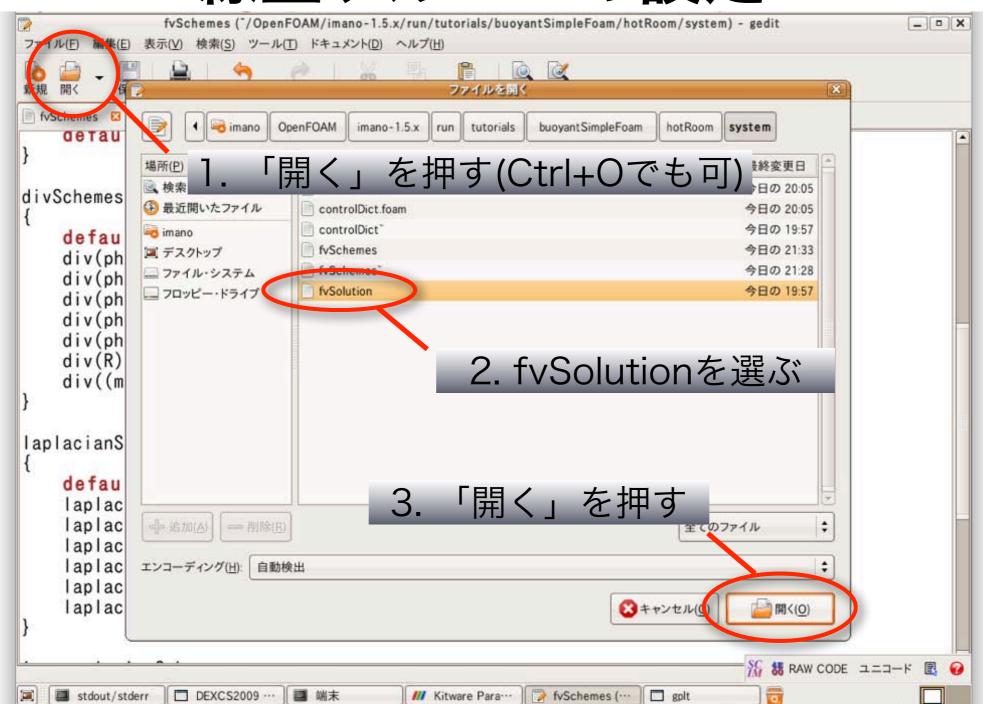
- ・計算は非常に安定
- •粗い格子では精度は悪い
- ・解が拡散的になる
- ・計算は不安定
- ・細かい格子では精度良い
- ・粗い格子では空間振動

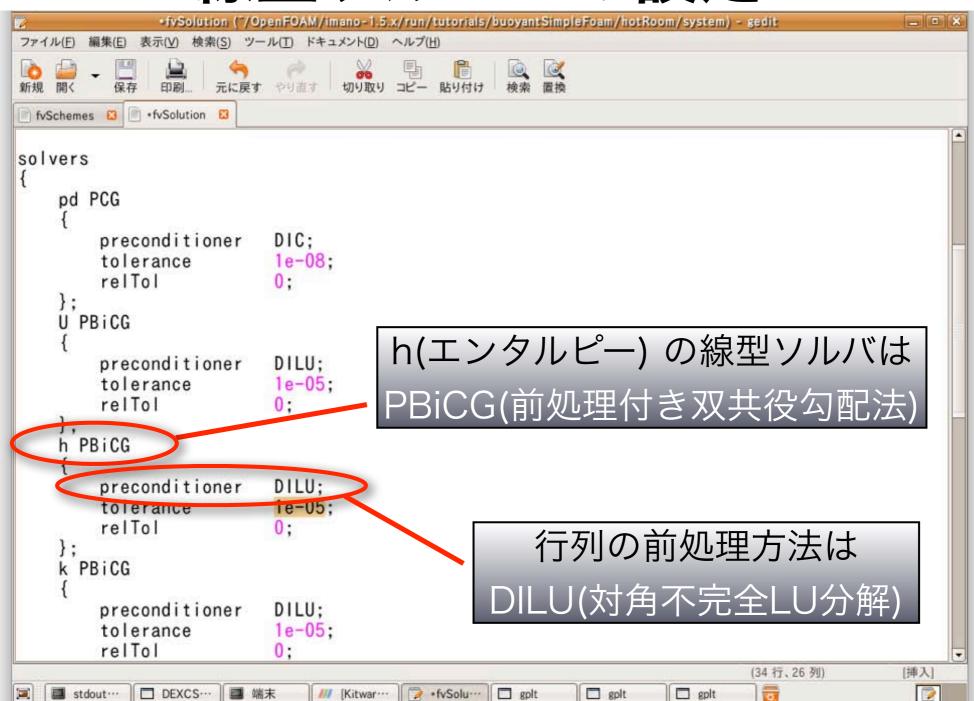
- ・計算は安定
- ・精度は程々良い
- ・引数を取る(パラメータ依存)

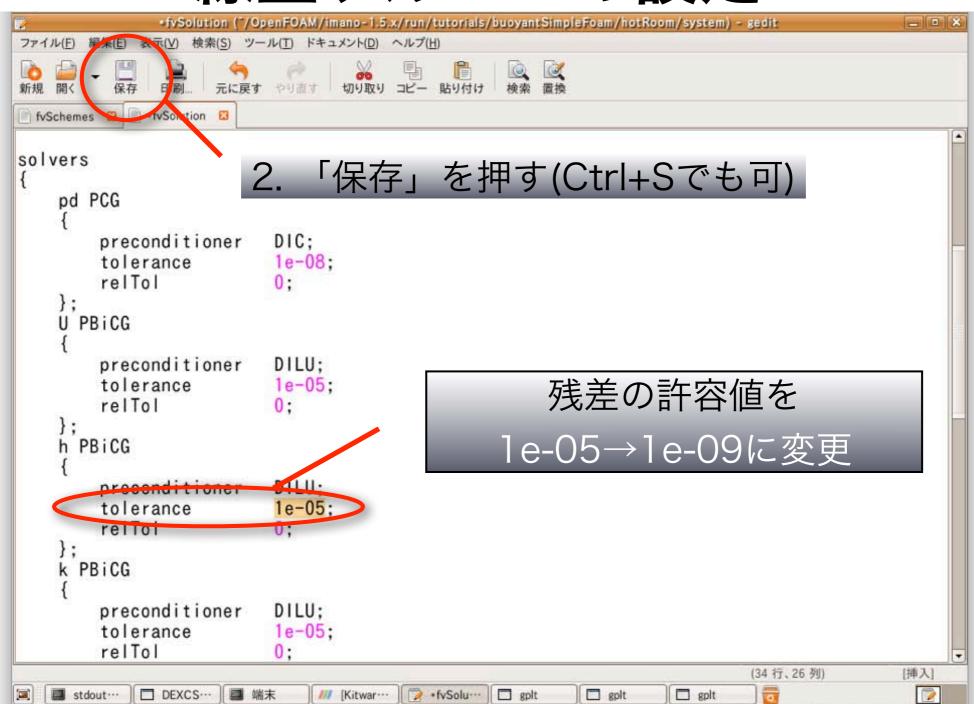
Tips その1

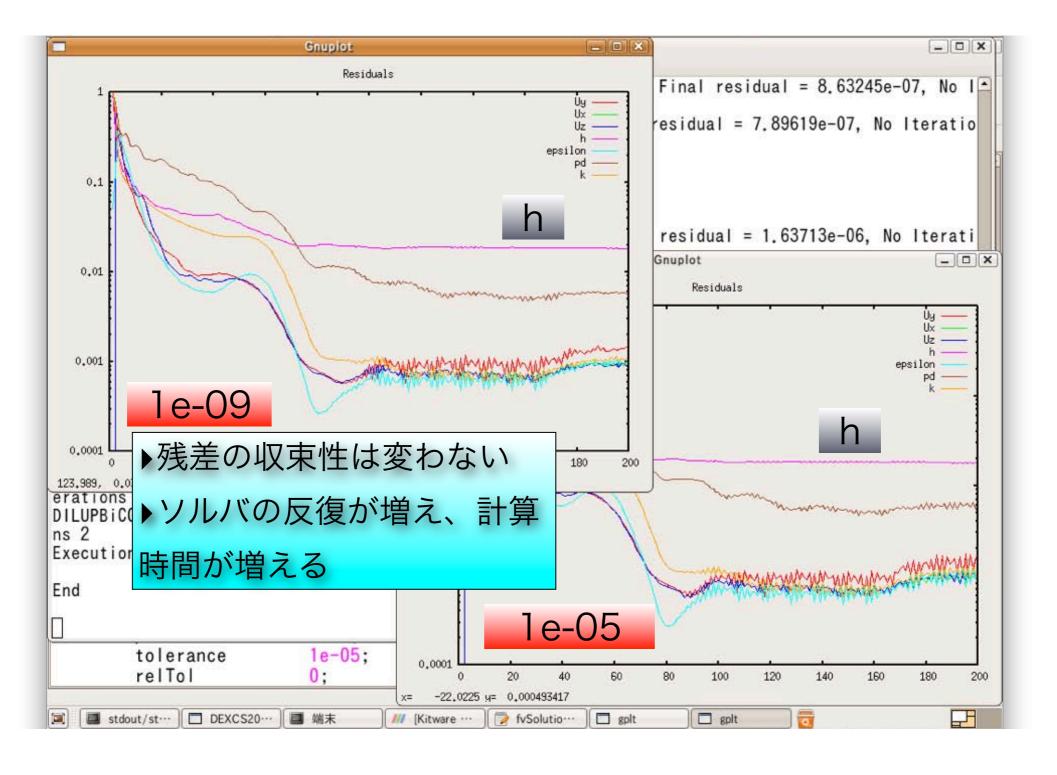
- 1. system/fvSchemesで、わざとスキーム名を間違えて書いてソルバーを実行すると、エラーに有効なスキーム一覧が出力される。
- 2. srcコマンドでソースの場所に行く
- 3. さらに、find . -name upwind でupwindスキームのあるディレクトリの場所がわかる。
- 4. ディレクトリ内の*.Cや*.Hがソース。マニュアルに詳細が書いていない場合にはソースを見てみよう。

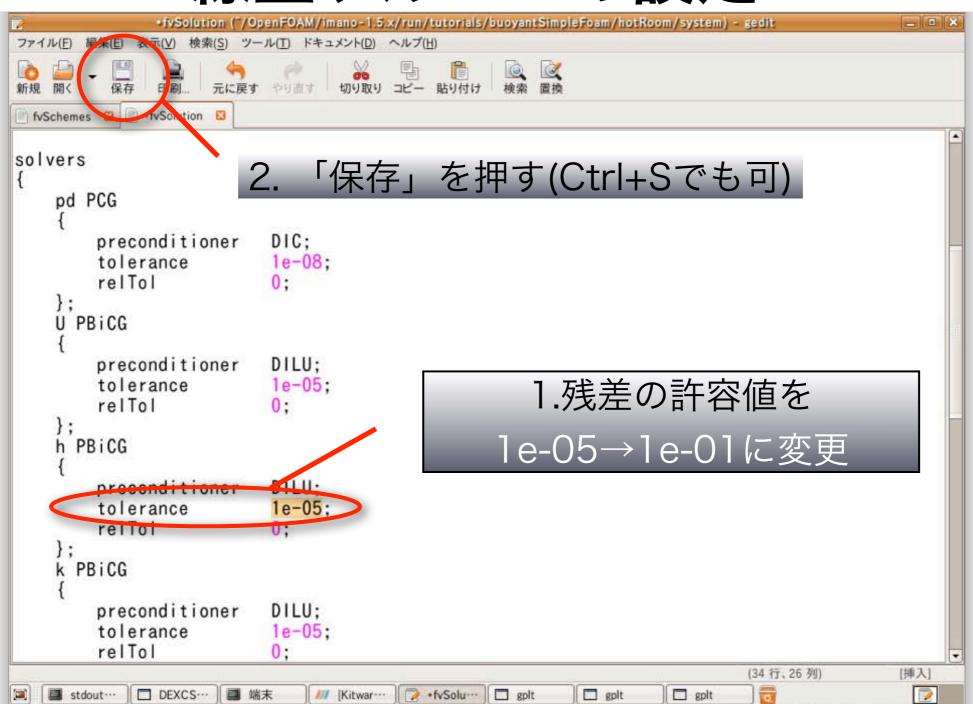
線型ソルバーの設定変更

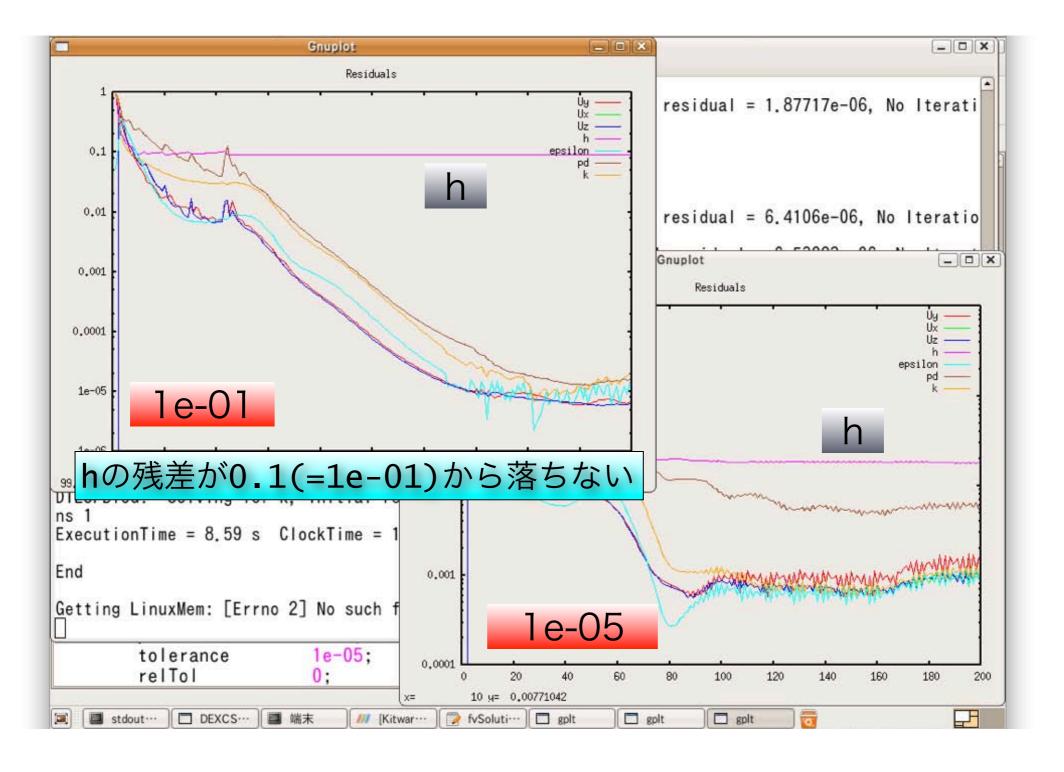


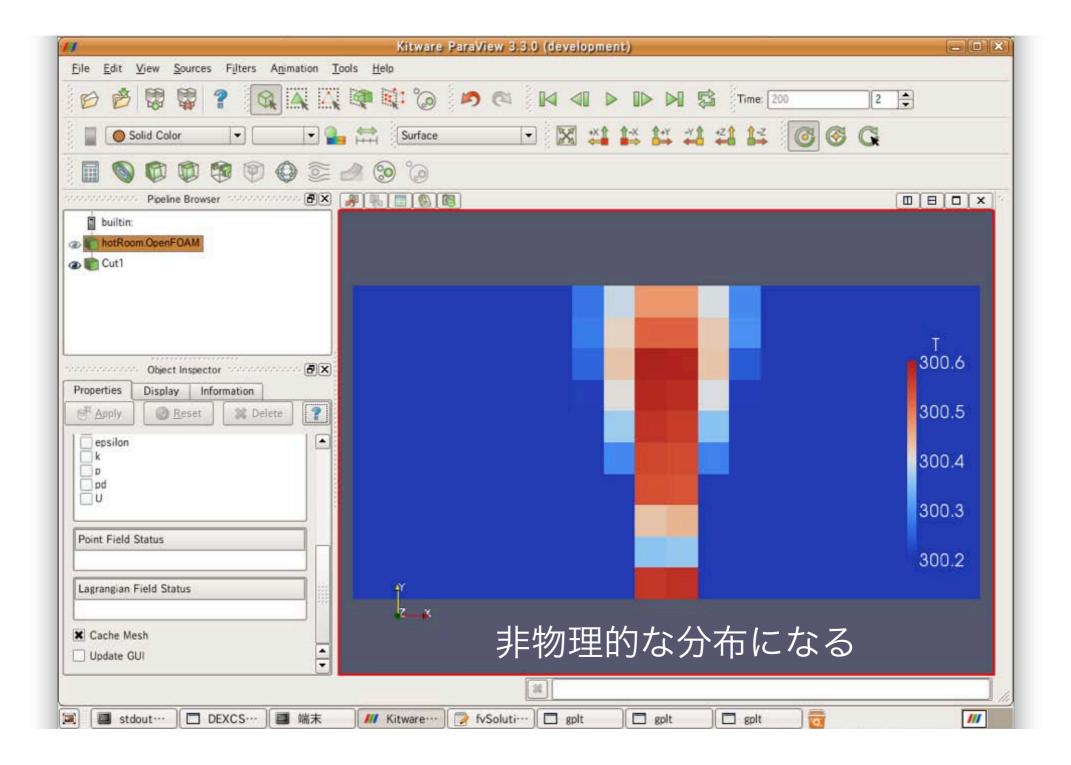


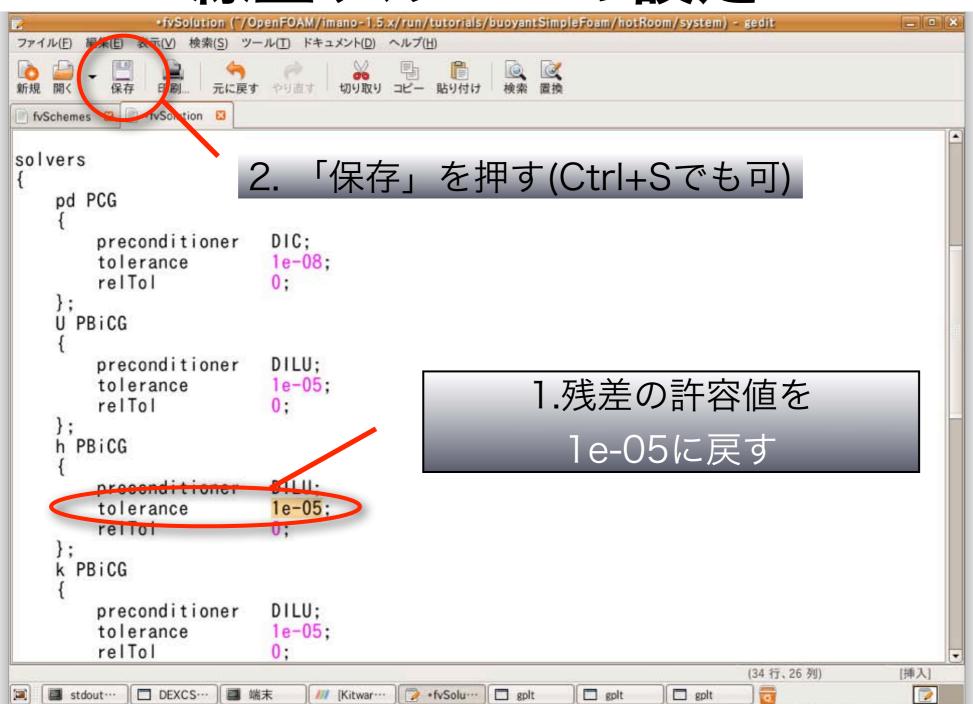










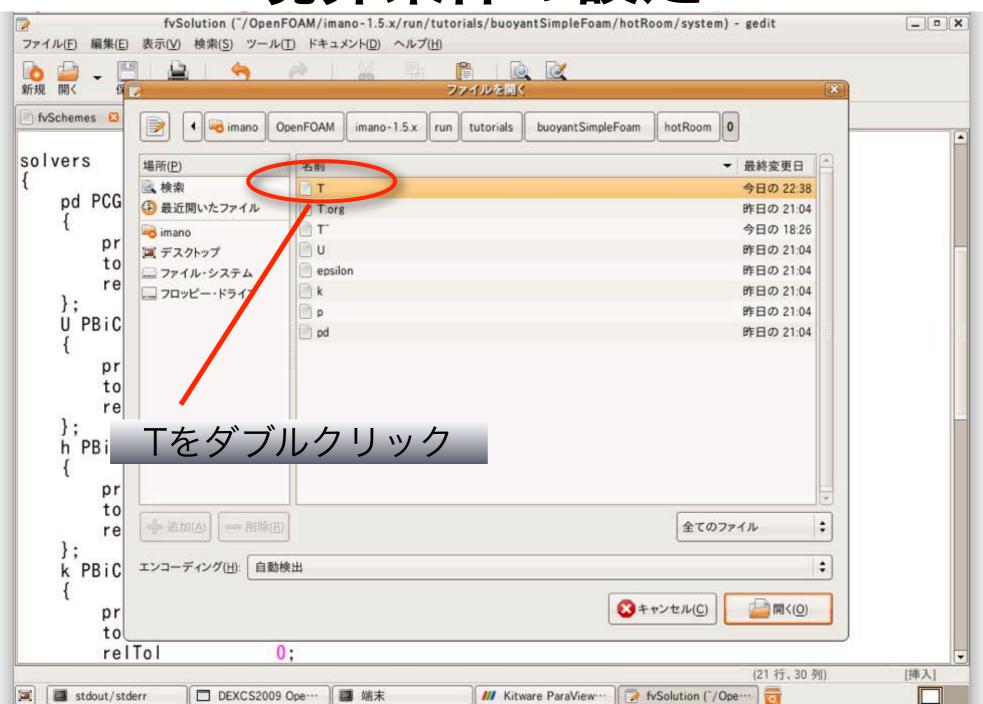


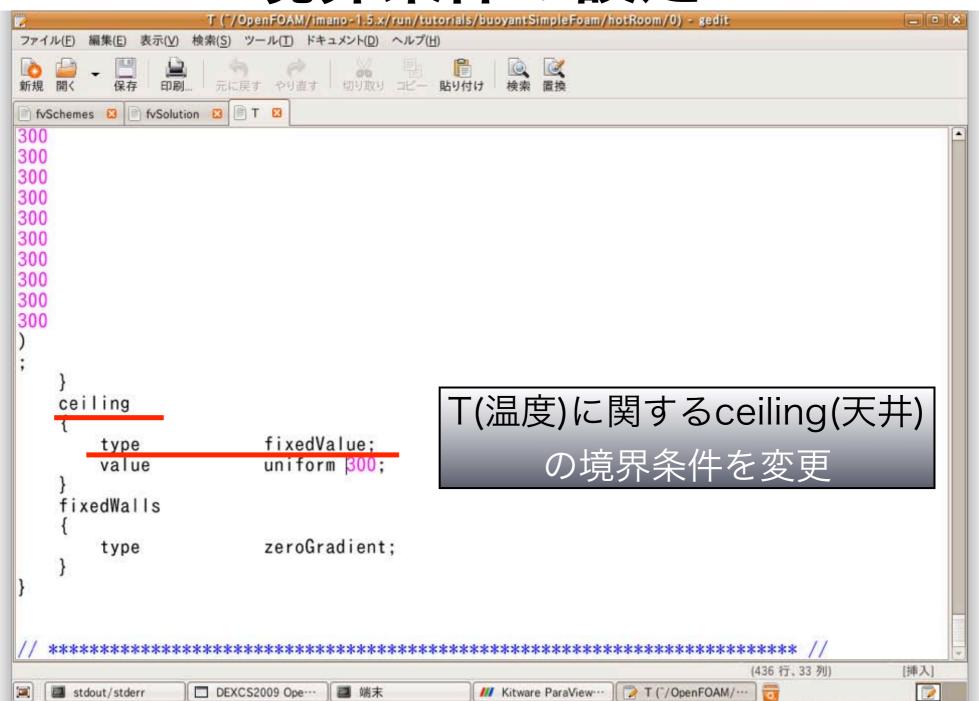
Tips その2

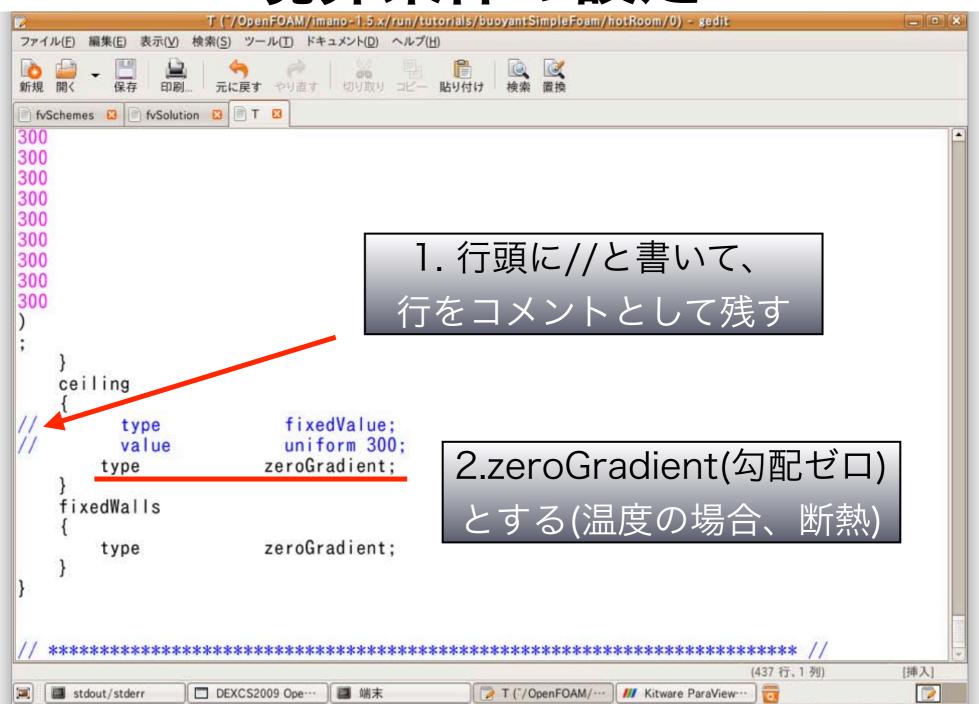
- 1. psコマンドでプロセス一覧が出る。
- 2. kill PID(番号) でプロセスが殺せる。
- 3. killall プロセス名で、そのプロセスが全て殺せる。
- 4. 例えば、killall gnuplot_x11とする
- と、pyFoamPlotRunner.py の実行で
- 残ったgnuplotのグラフが全て消える。
- 5. 複数のソルバーの実行を全て止めたい 時等にも killall が使える。

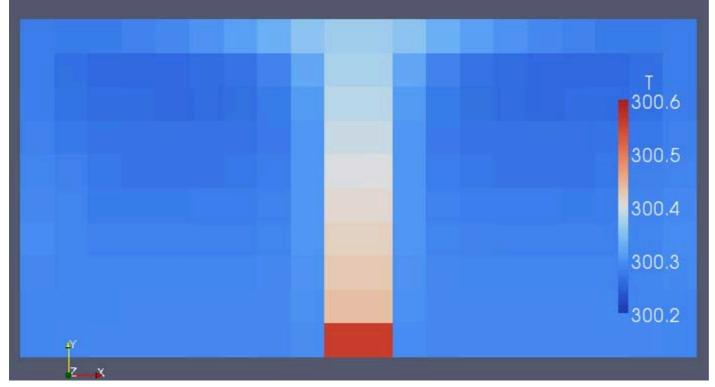
境界条件の設定変更





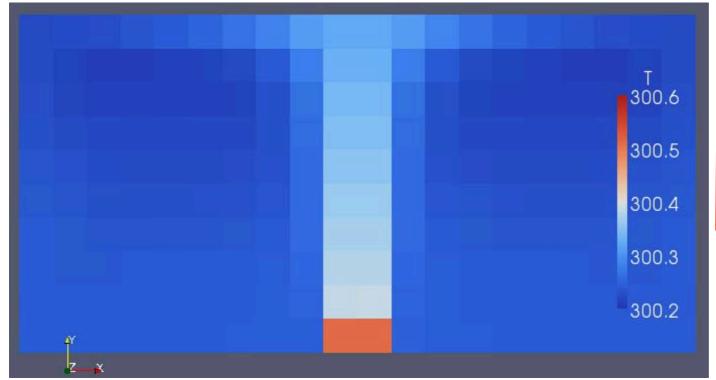




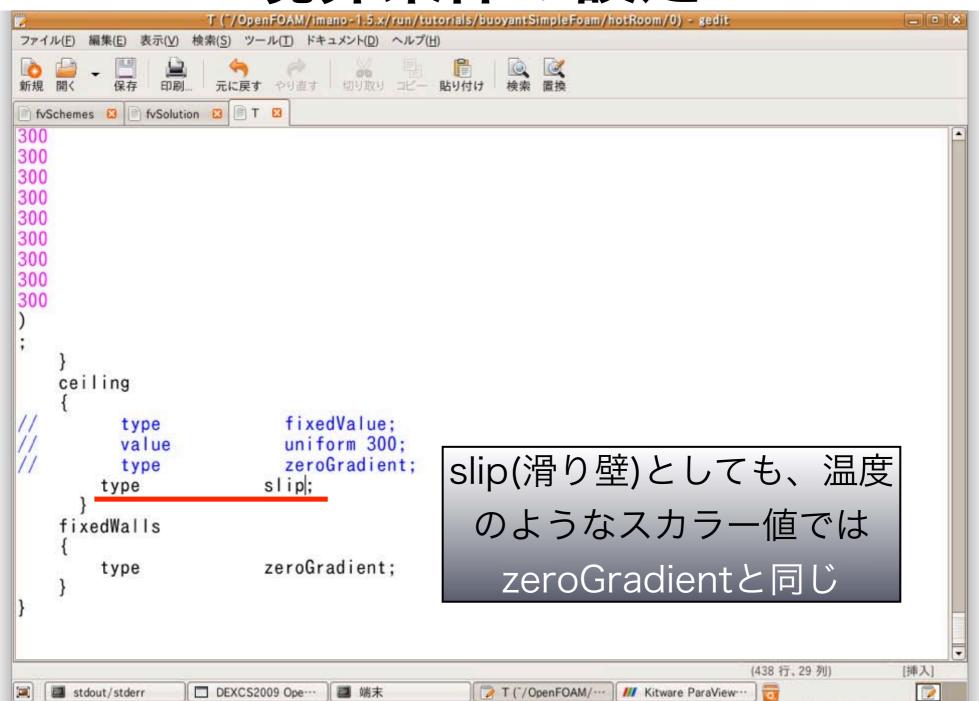


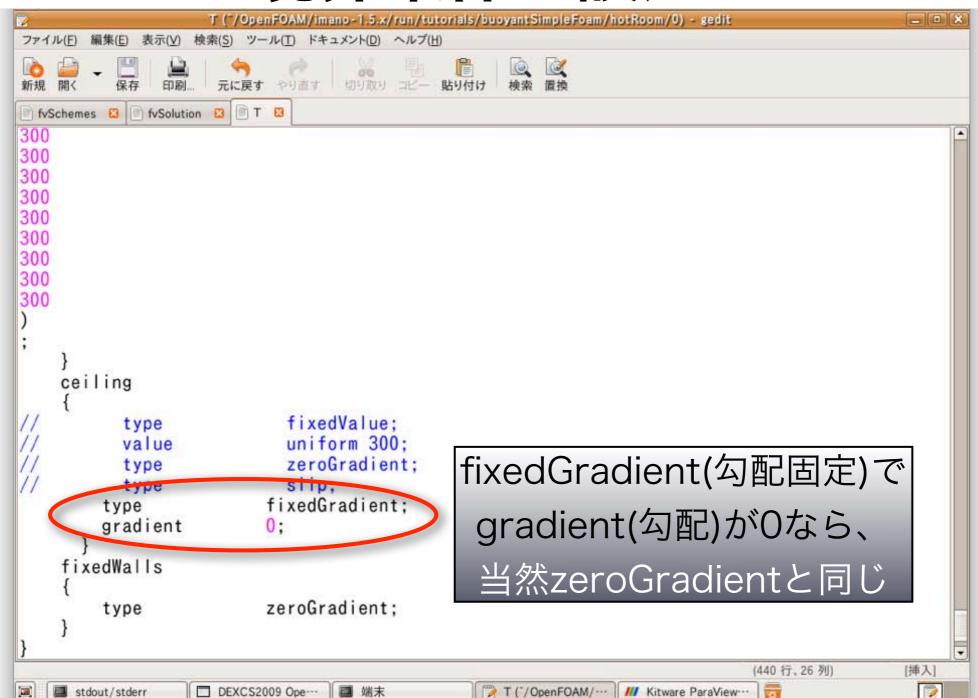
type zeroGradient;

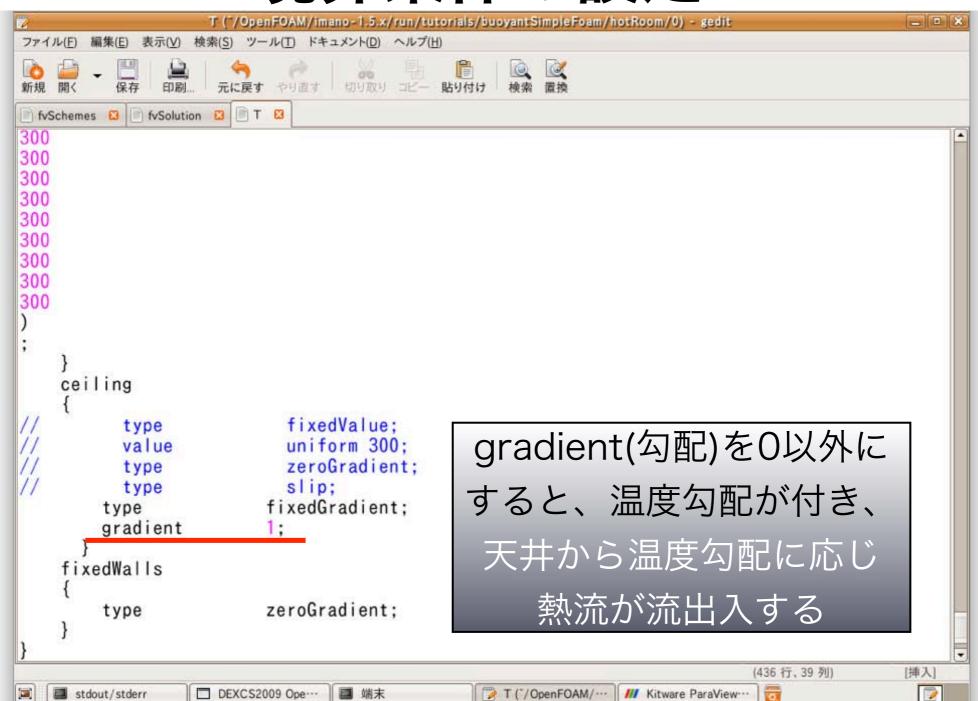
天井が断熱されて 全体的に温度が上昇

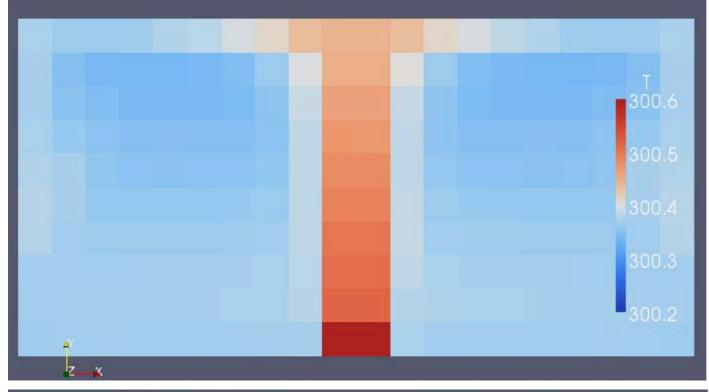


type fixedValue; value uniform 300;



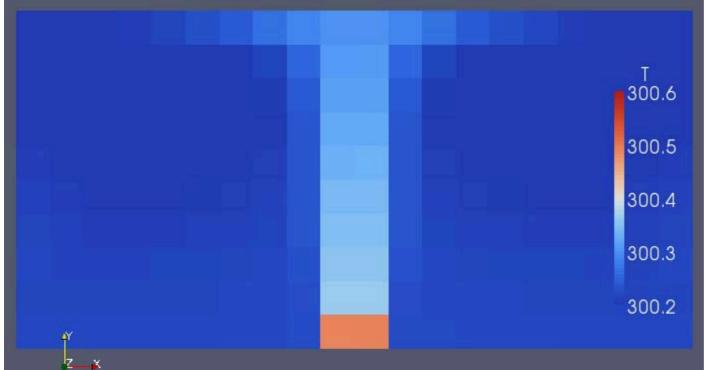






type fixedGradient;
gradient 1;

天井から熱が流入し 全体的に温度が上昇



type fixedGradient;
gradient -1;

天井から熱が流出し 全体的に温度が下降

Tips その3

- 1. src してから、find . -name fixed Value で、fixedValueの境界条件のソースがあるディ レクトリ ./finiteVolume/fields/ fvPatchFields/basic/fixedValue が出る。 2. cd ./finiteVolume/fields/fvPatchFields/ basic/ でbasicな境界条件のディレクトリに行 く。Is で境界条件一覧が見れる。
- 3. cd ../derived で複雑な境界条件のディレクトリに行く。Is で境界条件一覧が見れる。

Tips その4

- 1. tut でチュートリアルの場所に行く。
- 2. find . -name fvSolution | xargs grep GAMG で、線型ソルバー GAMG の設定をしている fvSolution の場所がわかる。
- 3.find . -name U | xargs grep time Varying 等とすれば、時刻ステップと共に変化する境界 条件を設定しているUの場所がわかる。

目次

- 1.下準備
- 2. 離散化スキームの設定変更
- 3. 線型ソルバーの設定変更
- 4. 境界条件の設定変更

今後の講習(希望あれば)

- 1. snappyHexMeshによる格子生成
- 2. faceSet等を用いた境界の設定
- 3. 並列計算方法
- 4. ソルバーのカスタマイズ
- 5. 乱流モデルのカスタマイズ