

平成22年度オープンCAE夏季臨時講習会(東京)

2010年8月6日

平成22年度オープンCAE夏季講習会(岐阜)

2010年8月8日

吹出口、吸込口、発熱条件の設定をした 室内の非等温流れ場解析

今野 雅 (オープンCAE学会、東京大学)



目次

1. 下準備

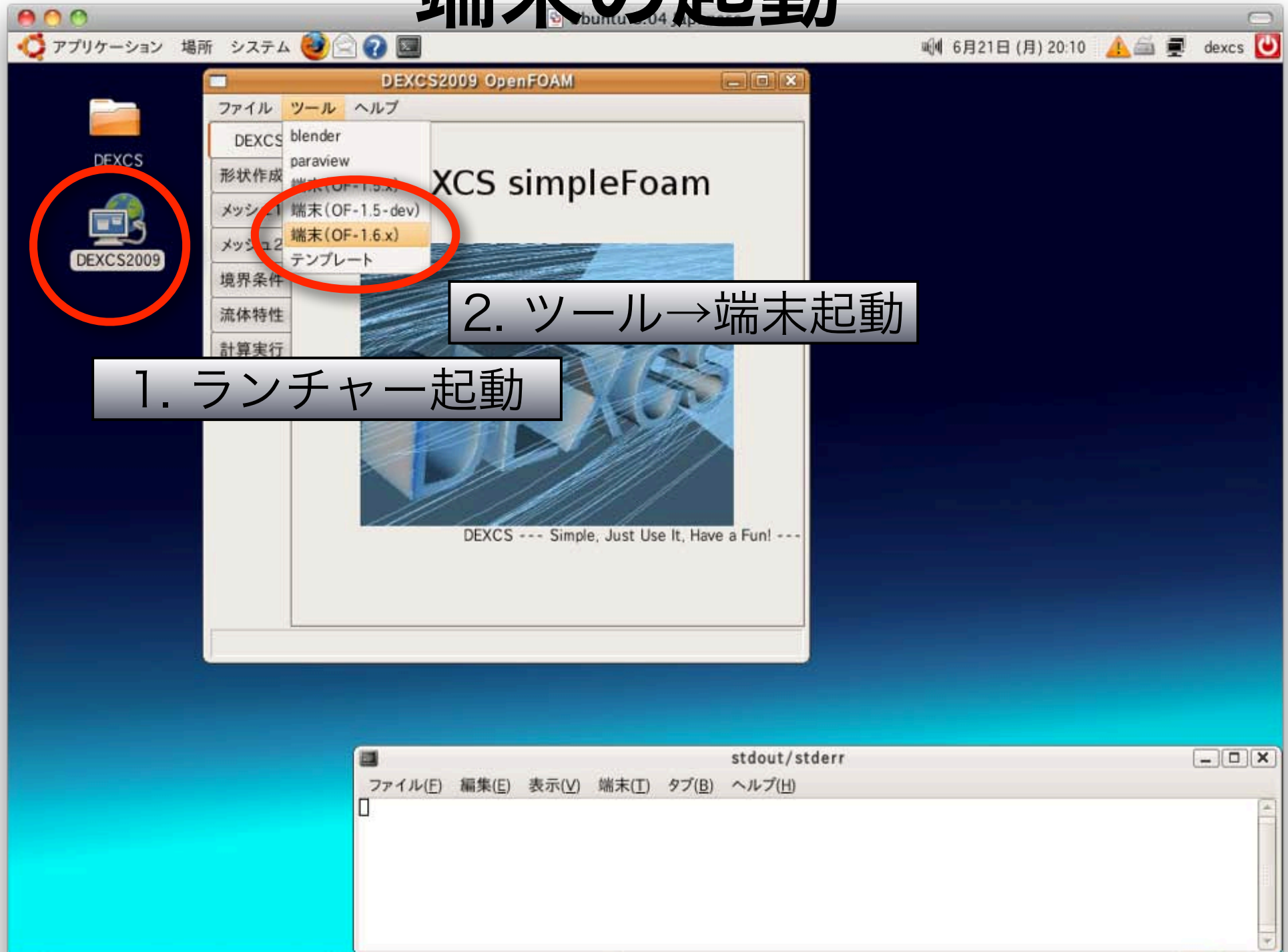
2. 発熱面に対応する新たなパッチ
作成と初期値の設定

3. 吹出口・吸込口に対応する新たな
パッチ作成と初期値の設定

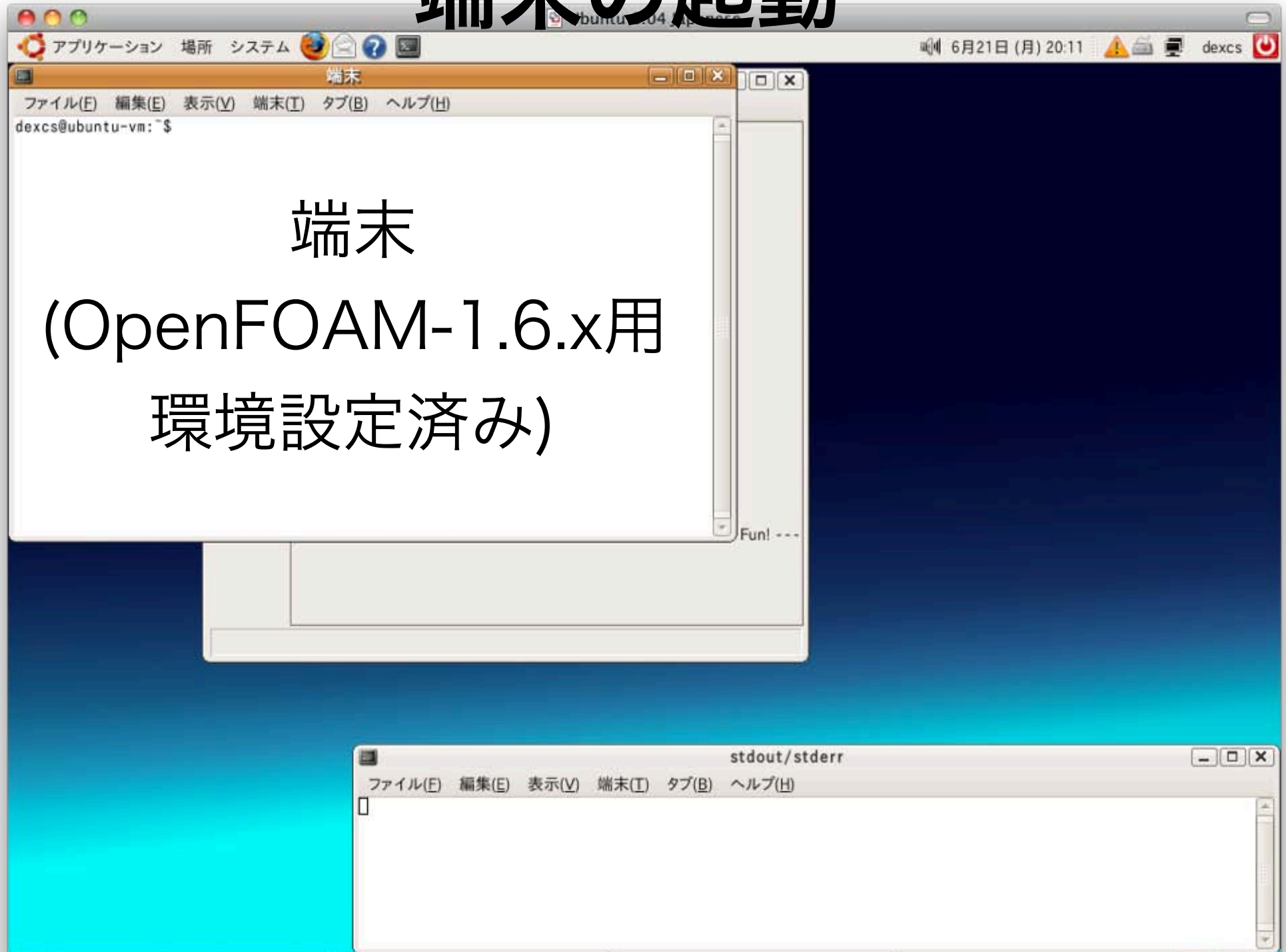
A stylized graphic of an atomic symbol, featuring a central green circle, two intersecting elliptical orbits (one light blue, one light green), and a small blue sphere representing an electron in the upper right orbit.

下準備

端末の起動



端末の起動



チュートリアルのコピー

まだコピーをしていない場合には、
端末で赤字のように打ってください

```
mkdir -p $FOAM_RUN↵
```

```
cp -r $FOAM_TUTORIALS $FOAM_RUN↵
```

チュートリアル場所に行く

```
run ↵
```

```
cd tutorials/ ↵
```

```
cd heatTransfer/ ↵
```

```
cd buoyantBoussinesqSimpleFoam/ ↵
```

```
cd hotRoom/ ↵
```

```
ls ↵
```

←ディレクトリ名はTab
キーで補完できます

出力

```
0 Allclean Allrun constant system
```

チュートリアルの実行

foamRunTutorials ↩

←Tabキーで補完できます

Allrunがあるチュートリアル・ケースにおいては、上記は Allrun を実行するのと同等です。

Allrunが無いケースでは、blockMeshと、そのケースに対応するソルバーが順次実行されます。

実行結果の全消去

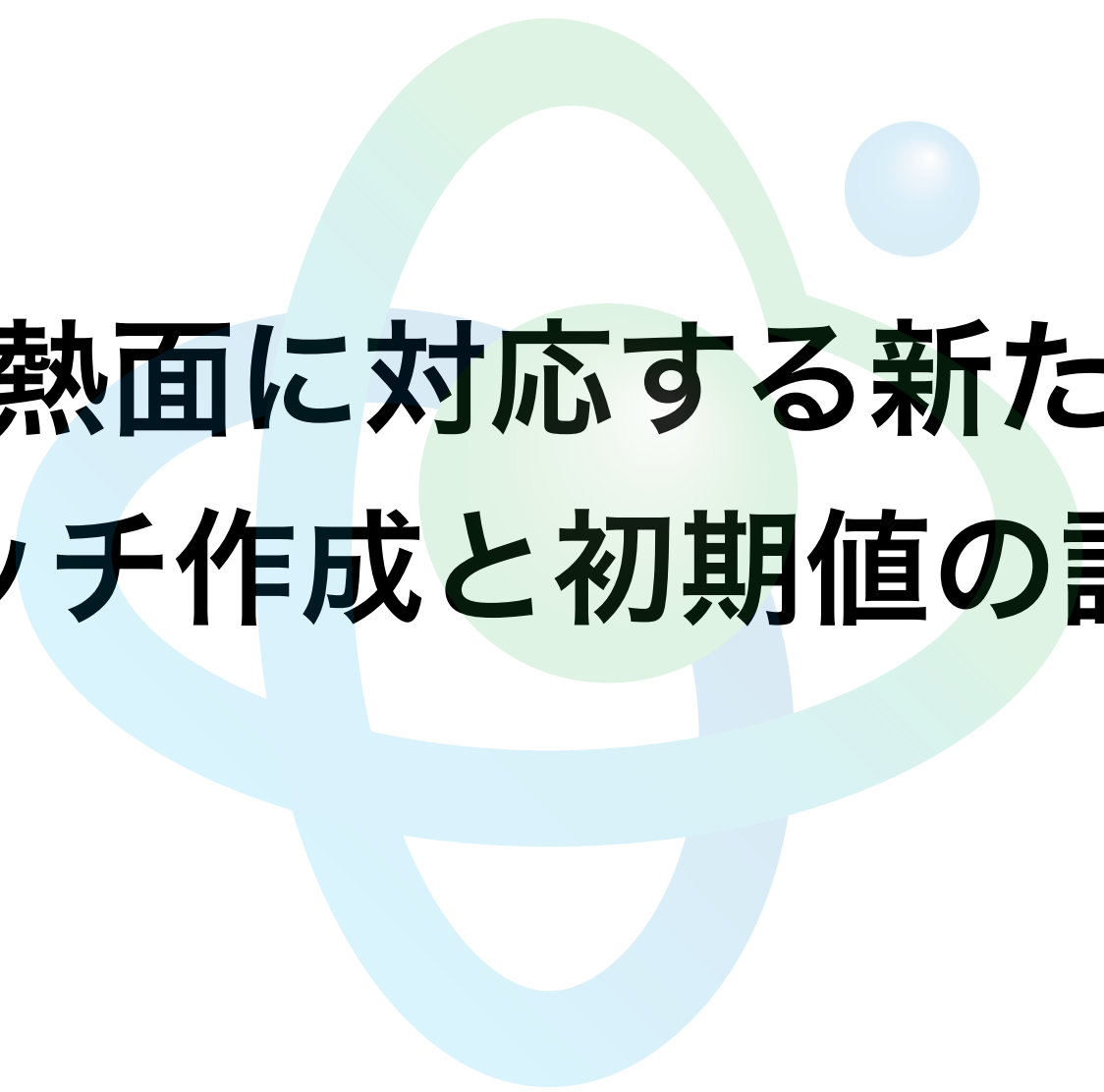
foamCleanTutorials ↩

←Tabキーで補完できます

出力

```
0 Allclean Allrun constant system
```

Allcleanがあるチュートリアル・ケースにおいては、
上記は ./Allclean を実行するのと同様です。



発熱面に対応する新たな パッチ作成と初期値の設定

新たなパッチ作成と境界条件設定

1. ベース格子の生成と確認

2. faceSetユーティリティで境界面にする境界の集合(faceSet)を作成

3. faceSetからcreatePatchユーティリティを使い、新たなpatchを作成

新たなパッチ作成と境界条件設定

4. 新たなpatchに対する境界条件を、各物理量の初期ファイルで設定

5. 新たなパッチを含んだ新格子の確認

6. 計算の実行

1. ベース格子生成と確認

blockMesh ↩

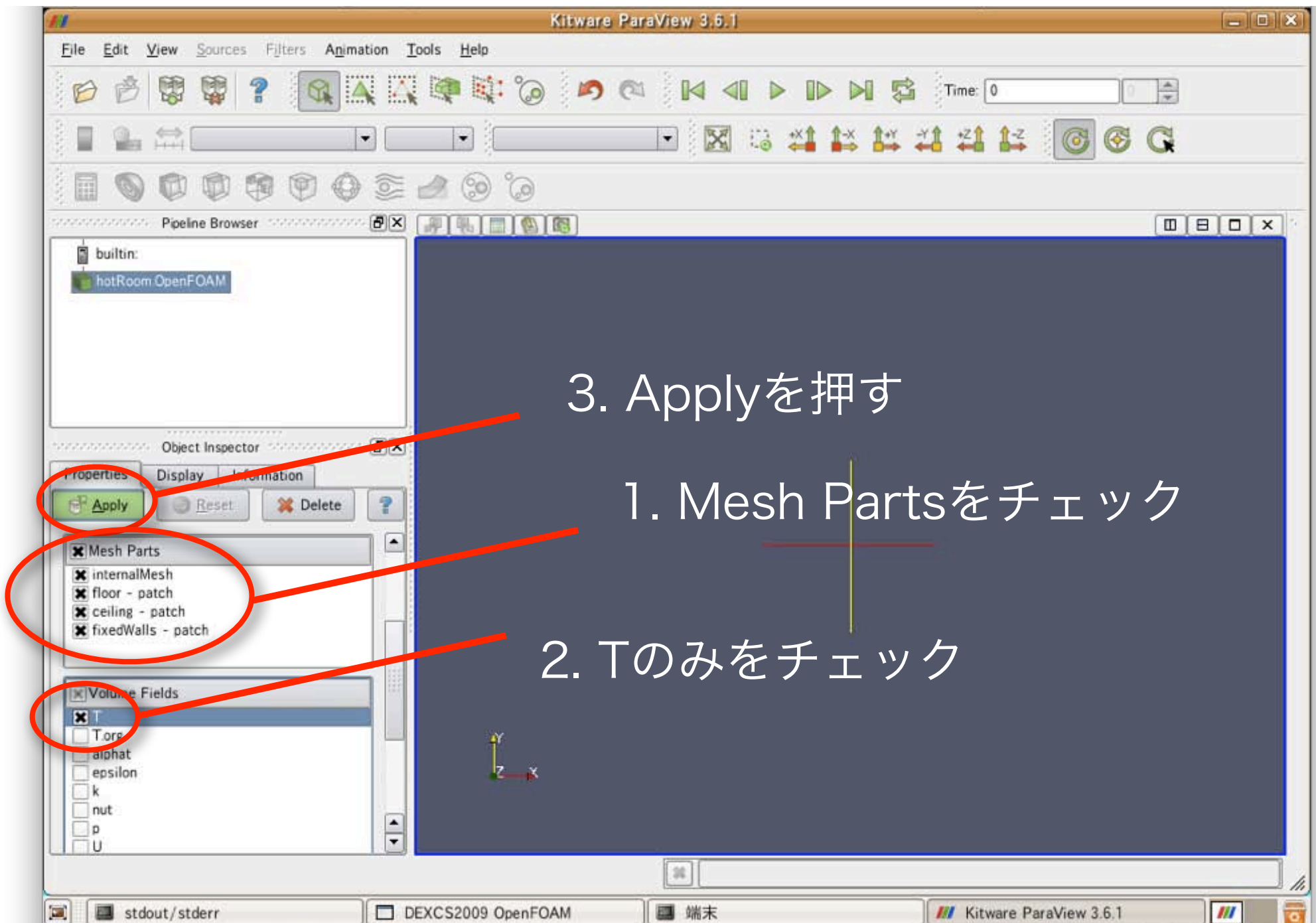
←ベースの格子生成

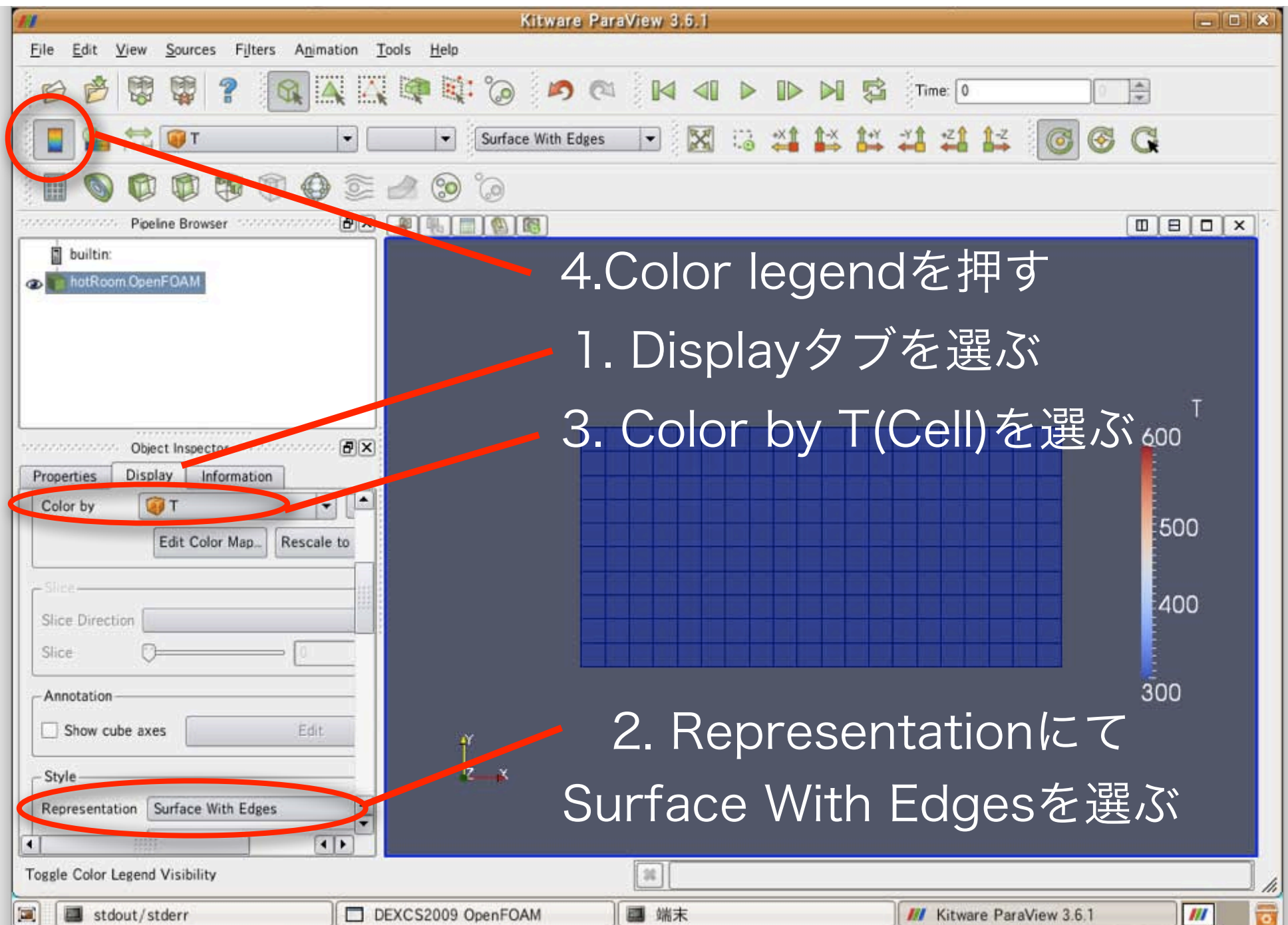
setHotRoom ↩

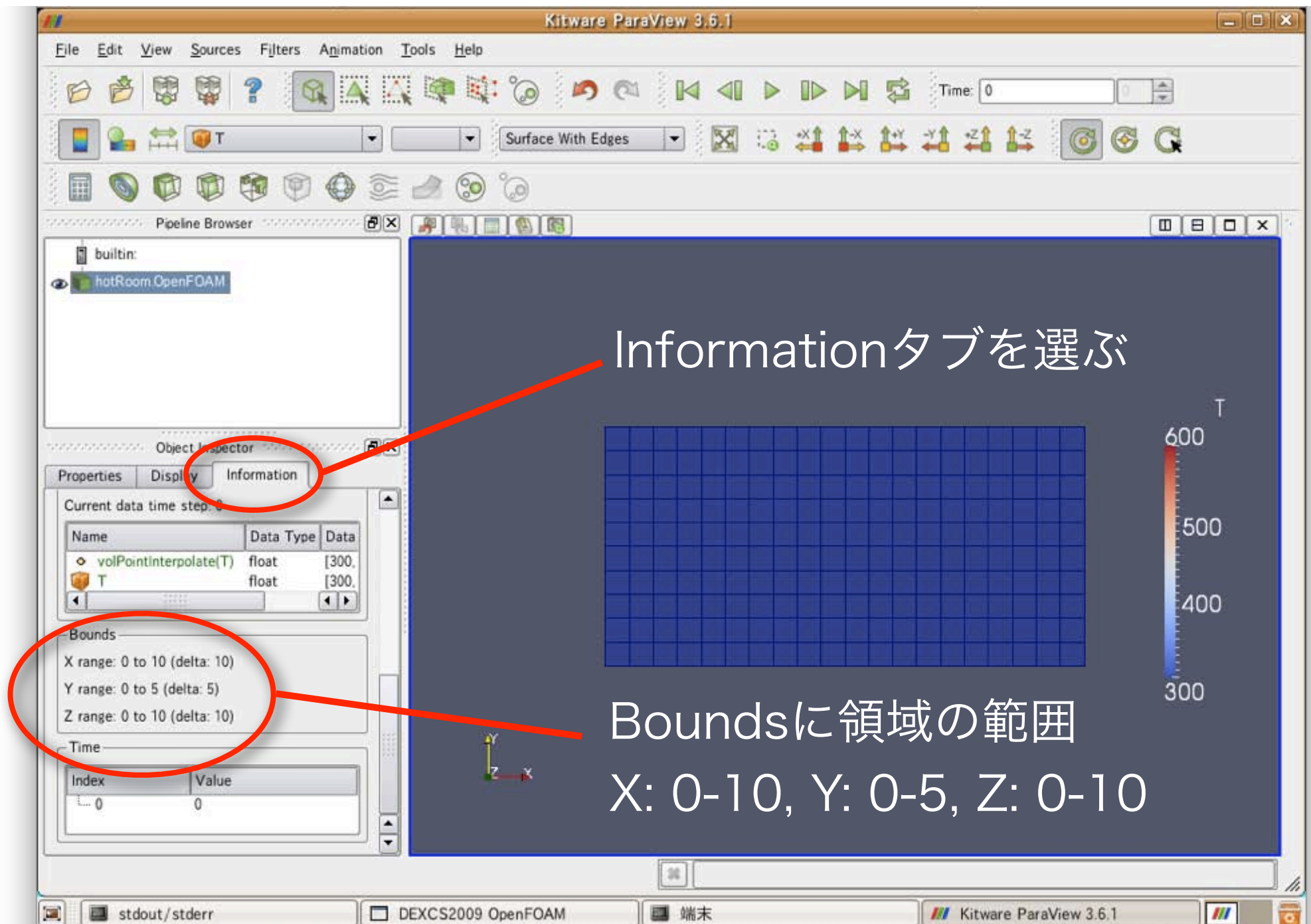
←発熱面の設定(専用Utility)

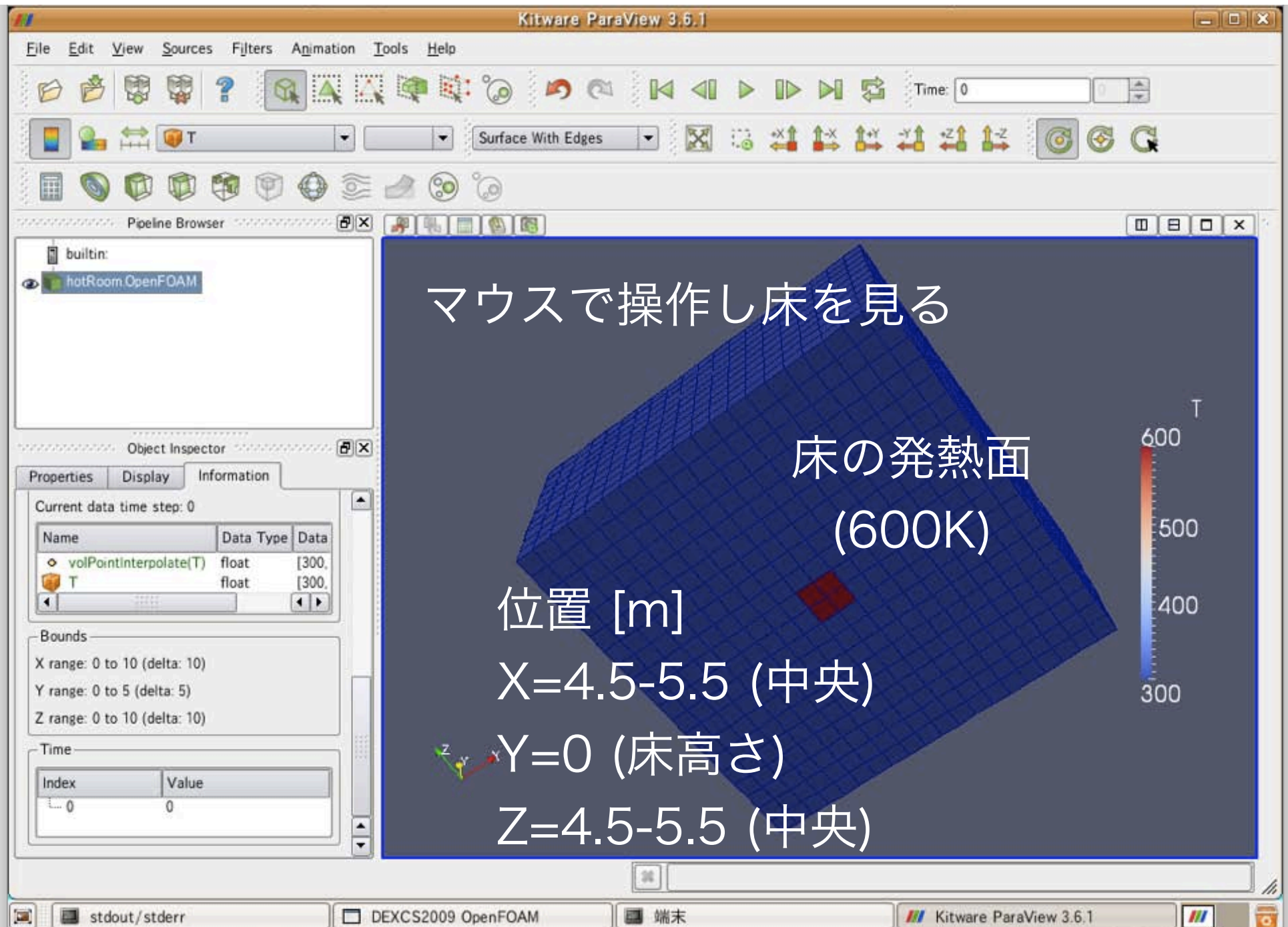
paraFoam & ↩

←可視化











2. faceSetユーティリティで境界
面にする境界の集合(faceSet)を作成

faceSetDict例のコピー

```
cd system↵
```

← *Dictはsystemに置く

```
find $FOAM_APP -name faceSetDict↵
```

出力

```
/home/dexcs/OpenFOAM/OpenFOAM-1.6.x/applications/utilities/  
mesh/manipulation/faceSet/faceSetDict
```

1. アプリケーションの設定ファイルのサンプルは、そのソースのディレクトリーに置いてあることが多い。
2. FOAM_APPは、アプリケーションのソースがあるディレクトリーを示す環境変数。

faceSetDict例のコピー

↓ 前回のコマンドを` (バッククォート) で囲む

```
cp `find $FOAM_APP -name faceSetDict` .
```

.(カレントディレクトリ)にコピー ↑

```
ls
```

出力

```
controlDict  faceSetDict  fvSchemes  fvSolution
```

` (バッククォート) で囲ったコマンドの出力が、外側のコマンドの引数になるので、"cp faceSetDictの場所 ." が実行され、faceSetDictがカレントディレクトリーにコピーされる。

createPatchDict例のコピー

```
cp `find $FOAM_APP -name createPatchDict` .  
ls
```

出力

```
controlDict  createPatchDict  faceSetDict  fvSchemes  
fvSolution
```

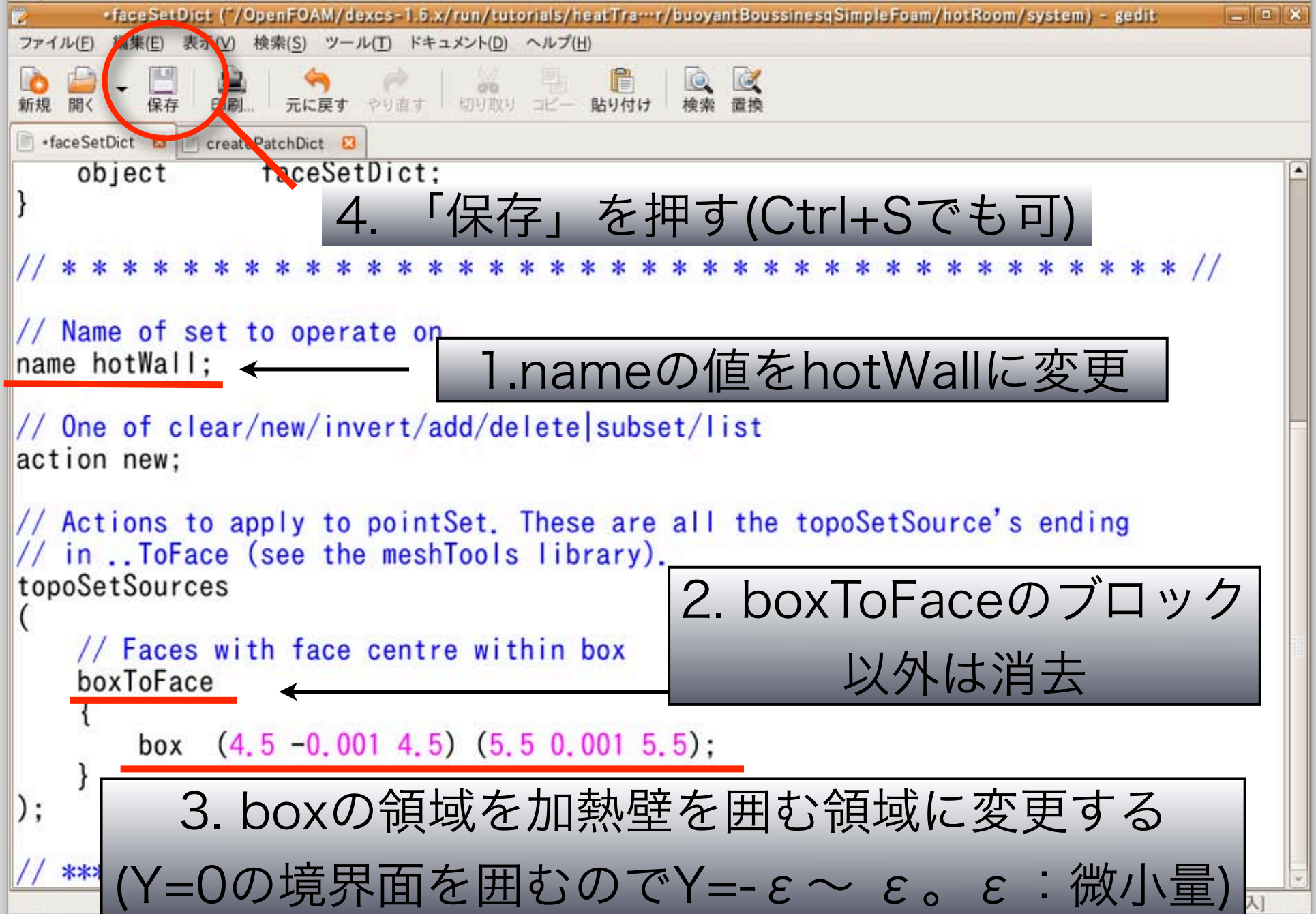
ファイラー等で手動で検索したり、コピーしたりする
のとは異なり、シェルのコマンドで検索&コピー等の操
作行くと、似たような処理を繰り返す場合に便利。

各種設定ファイルの修正

```
gedit faceSetDict createPatchDict &↵
```

1. geditで起動して、ファイル名を選んでも良いですが、最初から編集するファイル名を引数に指定すると早い。
2. 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わない。
3. 最後に&を付けてバックグラウンドでgeditを動かす。

faceSetDictの修正



4. 「保存」を押す(Ctrl+Sでも可)

1.nameの値をhotWallに変更

2. boxToFaceのブロック
以外は消去

3. boxの領域を加熱壁を囲む領域に変更する
(Y=0の境界面を囲むので $Y = -\varepsilon \sim \varepsilon$ 。 ε : 微小量)

faceSetの実行

```
cd ..
```

←ケースディレクトリに戻る

```
faceSet
```

出力

```
Reading faceSetDict
```

```
Backing up hotWall into hotWall_old
```

```
Set:hotWall Size:0 Action:new
```

```
    Adding faces with centre within box (4.5 -0.001 4.5)  
(5.5 0.001 5.5)
```

```
Writing hotWall (size 4) to "constant/polyMesh/sets/hotWall"
```

4つの格子界面のfaceSetが、constant/polyMesh/sets/hotWallとして書き出された。

hotWallのfaceSetの確認

[more constant/polyMesh/sets/hotWall](#) ↩

moreのキー/スペース：続ける、b：バック、q：終了

出力

```
4  
(  
11389  
11390  
11409  
11410  
)
```

faceSetの実際の中身は格子界面の番号のリスト
(通常、中身を詳しく見る必要はない。)



3. faceSetからcreatePatch
ユーティリティを使い、新たな
patchを作成

createPatchDictの修正

5. ファイル→終了(Ctrl+Qでも可)

1. createPatchDictのタブ選択

3. patch名(name)の値をhotFloorに変更

2. constructFrom setのブロック以外は消去

4. setの値を先程作成したfaceSet名のhotWallに変更

```
// Patches to create.  
patchInfo  
(  
  {  
    name hotFloor;  
  
    // Type of new patch  
    dictionary  
    {  
      type wall;  
    }  
  
    constructFrom set;  
    patches ();  
  
    set hotWall;  
  }  
);
```

```
// ***** //
```

(54 行、22 列)

[挿入]

createPatch

createPatch -overwrite ↩

↑ 通常は新格子が次の時刻ディレクトリに出力されるが、`overwrite`オプションで現格子を上書きする

出力

```
Adding new patch hotFloor as patch 3 from
{
    type                wall;
(snip)
Read 4 faces from faceSet hotWall
(snip)
Writing repatched mesh to 0
```

hotWallのfaceSetからなる新しいpatchである
hotFloorが、0(constant/polyMesh)に上書きで出力

格子の境界の確認

more constant/polyMesh/boundary ↩

スペース：続ける、b：バック、q：終了

出力

```
hotFloor
{
    type            wall;
    nFaces          4;
    startFace       12796;
}
```

格子の境界情報を表わすboundaryにhotFloorのpatchがtype wallとして追加されている。
(通常、中身を詳しく見る必要はない。)



4. 新たなpatchに対する境界条件
を、各物理量の初期ファイルで設定

各物理量の初期ファイルの修正

```
ls -al 0/  
gedit 0/* &↵
```

1. 各物理量の初期ファイルは通常0の下
2. ls -alで、ファイルのサイズがエディターで開けない程大きくないか確認する。
3. *(アスタリスク)は.(ドット)で始まるファイルやディレクトリ以外 (ワイルドカード) に置換され、全初期ファイルが編集対象となる。

温度の初期ファイル(0/T)の修正

1. Tのタブが選択されていることを確認

```
internalField    uniform 300;

boundaryField
{
    floor
    {
        type      fixedValue;
        value      uniform 300;
    }
    hotFloor
    {
        type      fixedValue;
        value      uniform 600;
    }
    ceiling
    {
        type      fixedValue;
        value      uniform 300;
    }
    fixedWalls
    {
        type      zeroGradient;
    }
}
```

2. floorのpatchのvalueはnonuniformから、uniform 300に変更

3. hotFloorのpatchを追加

4. valueはuniform 600

(30 行, 36 列) [挿入]

速度の初期ファイル(0/U)の修正

1. Uのタブ選択

```
internalField    uniform (0 0 0);

boundaryField
{
    floor
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }

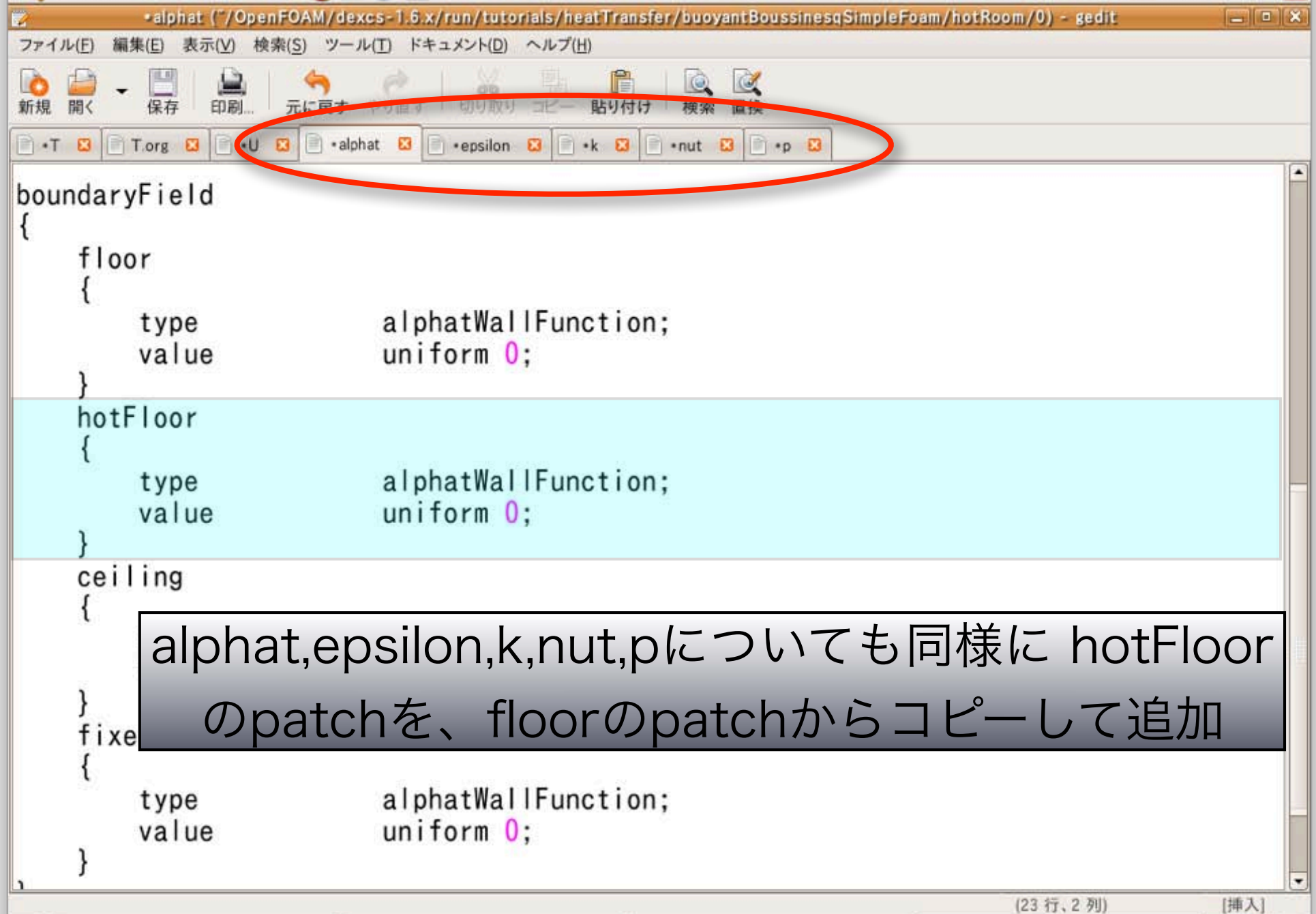
    hotFloor
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }

    ceiling
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }
}
```

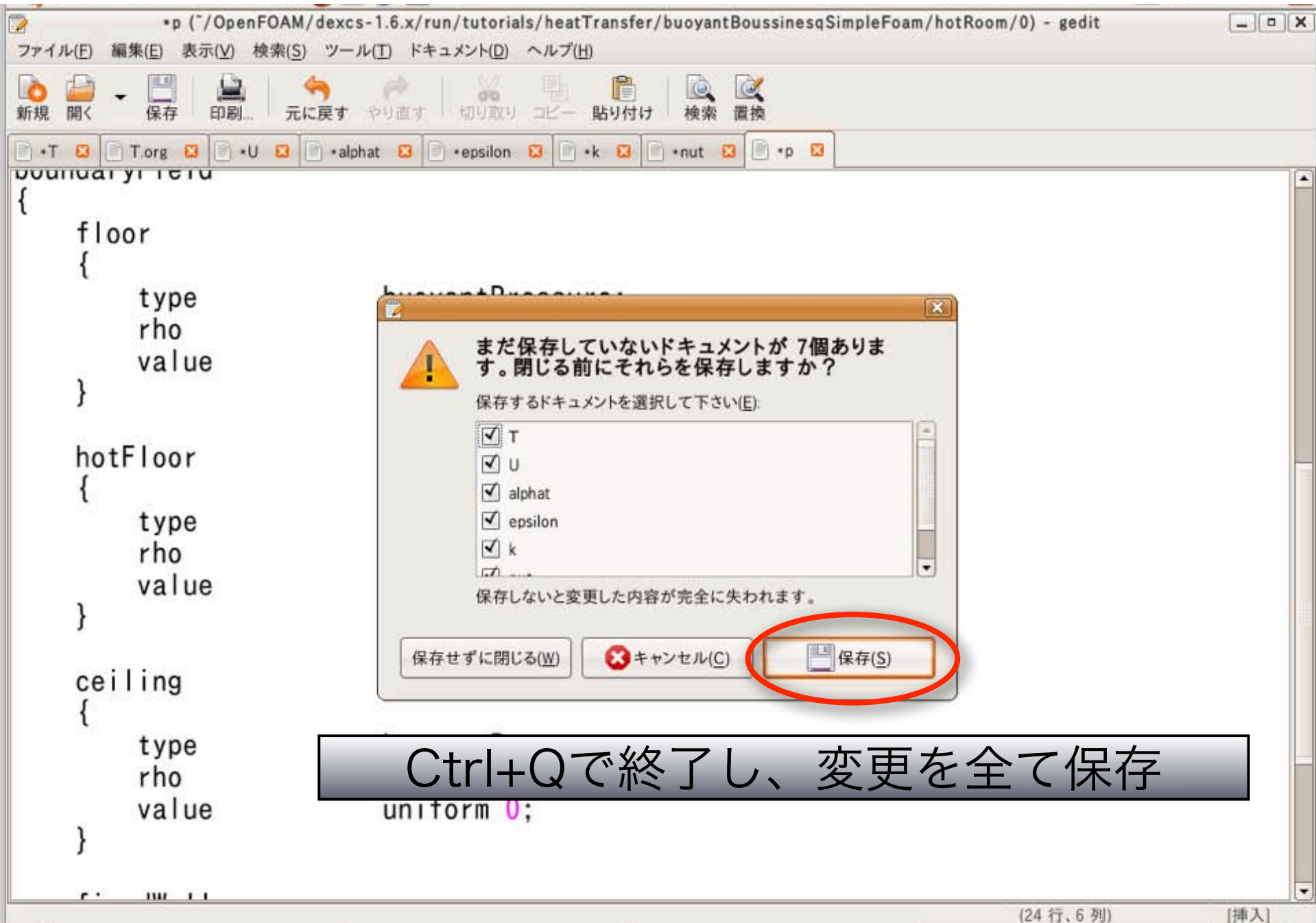
3. hotFloorのpatchを、floorのpatchからコピーして追加

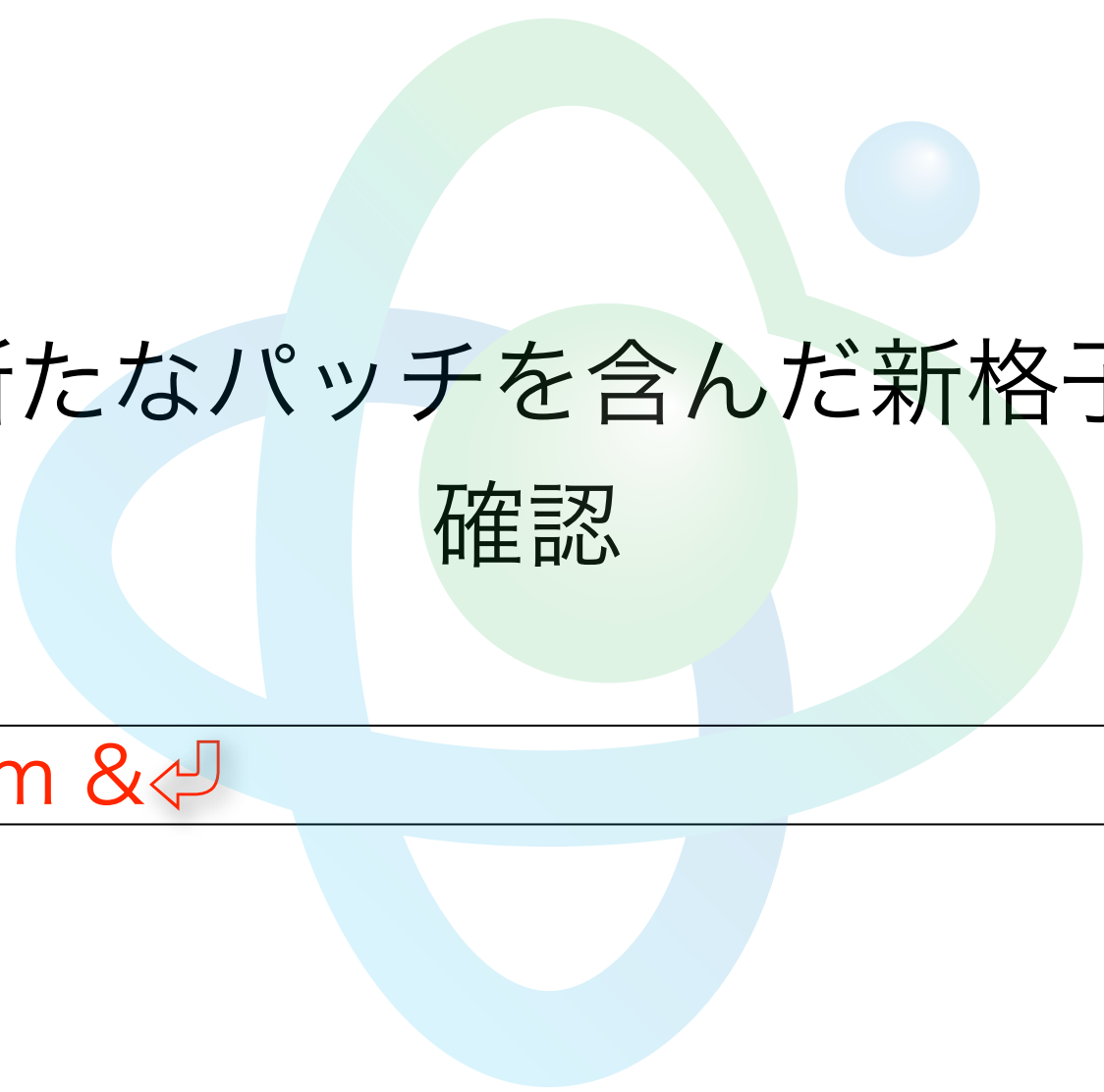
(31 行、36 列) [挿入]

その他の初期ファイルの修正



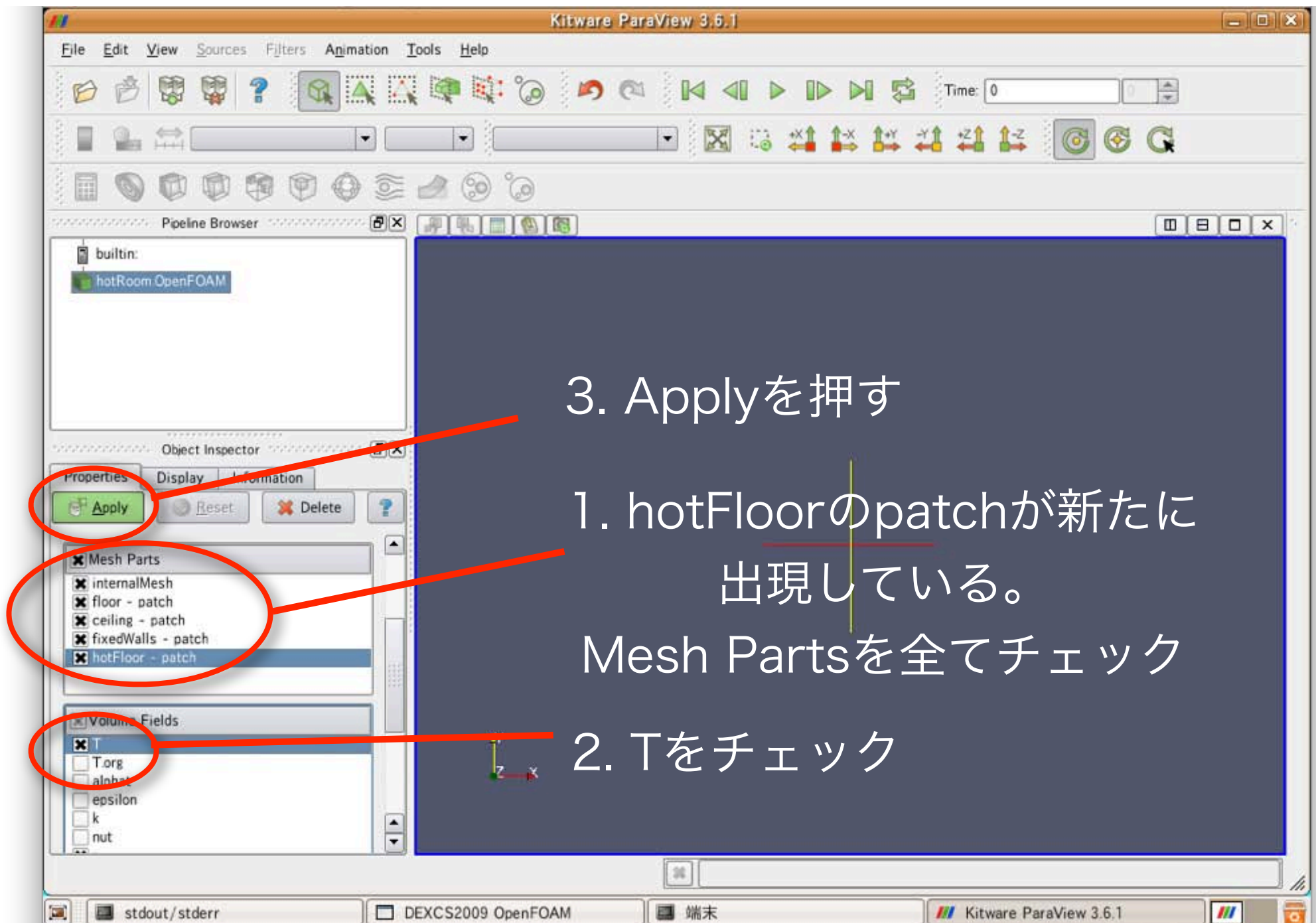
その他の初期ファイルの修正

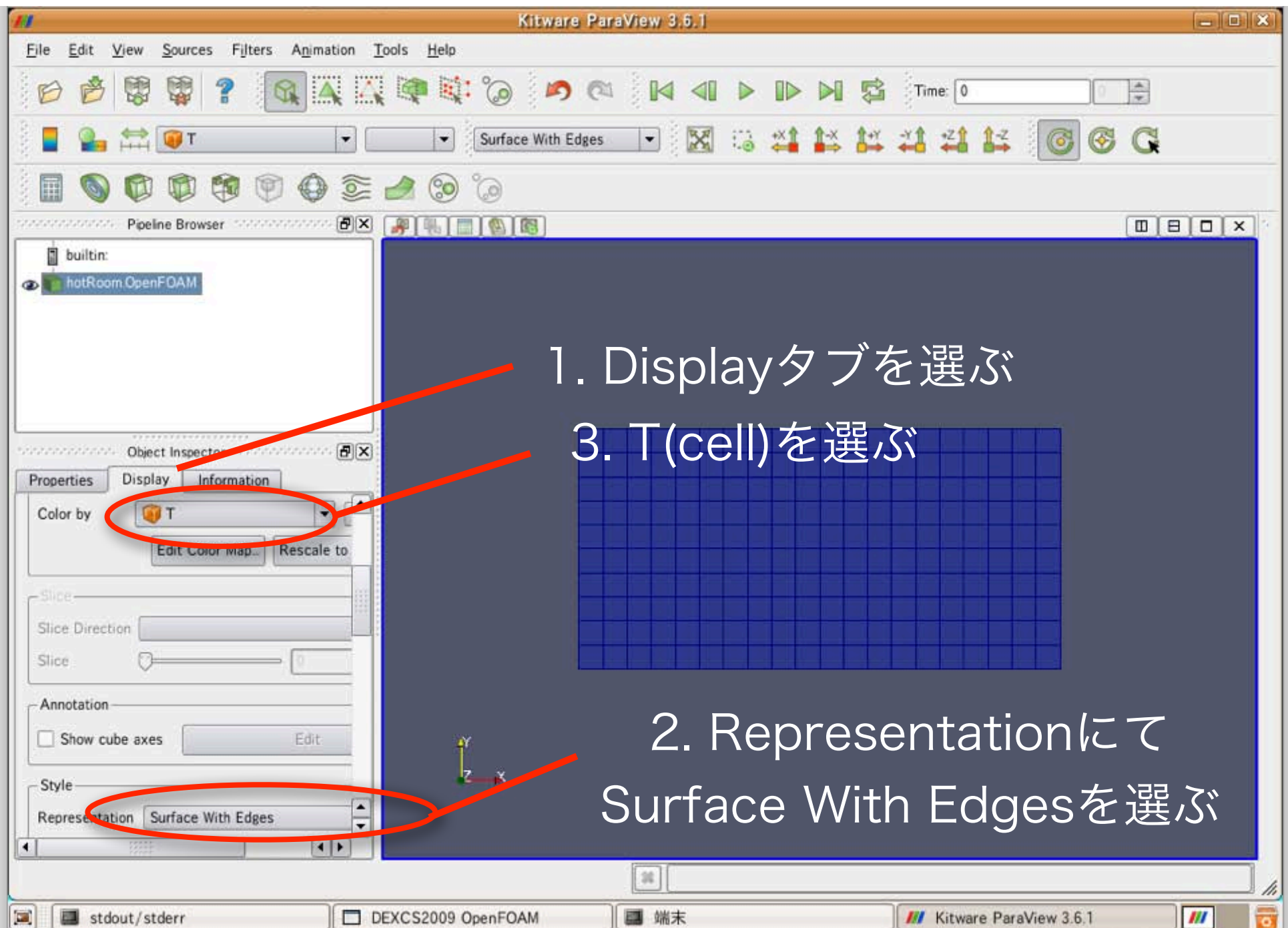


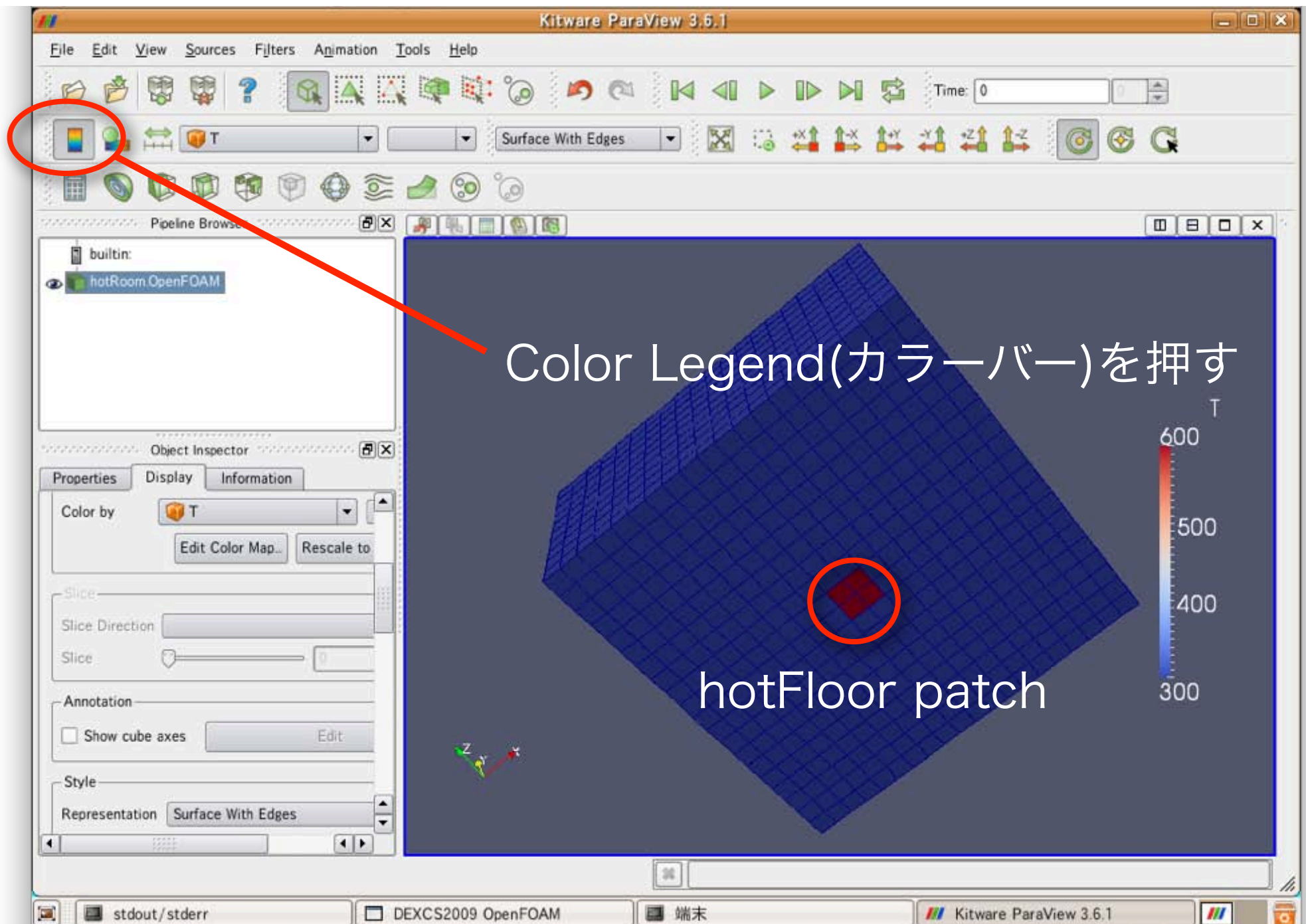


5. 新たなパッチを含んだ新格子の 確認

paraFoam & ↩







6. 計算の実行

buoyantBoussinesqSimpleFoam ↵

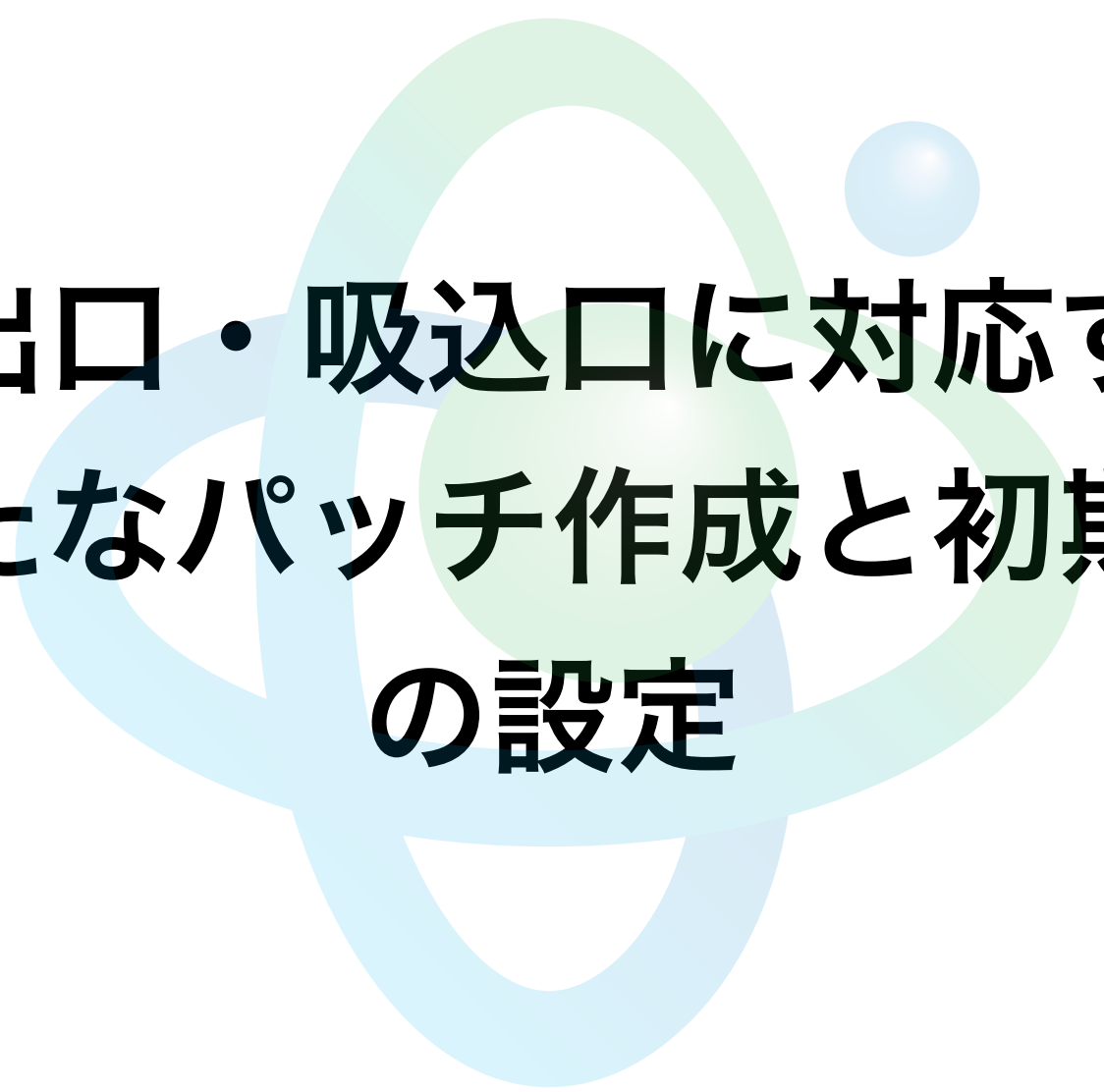
または

↓ logを残す場合

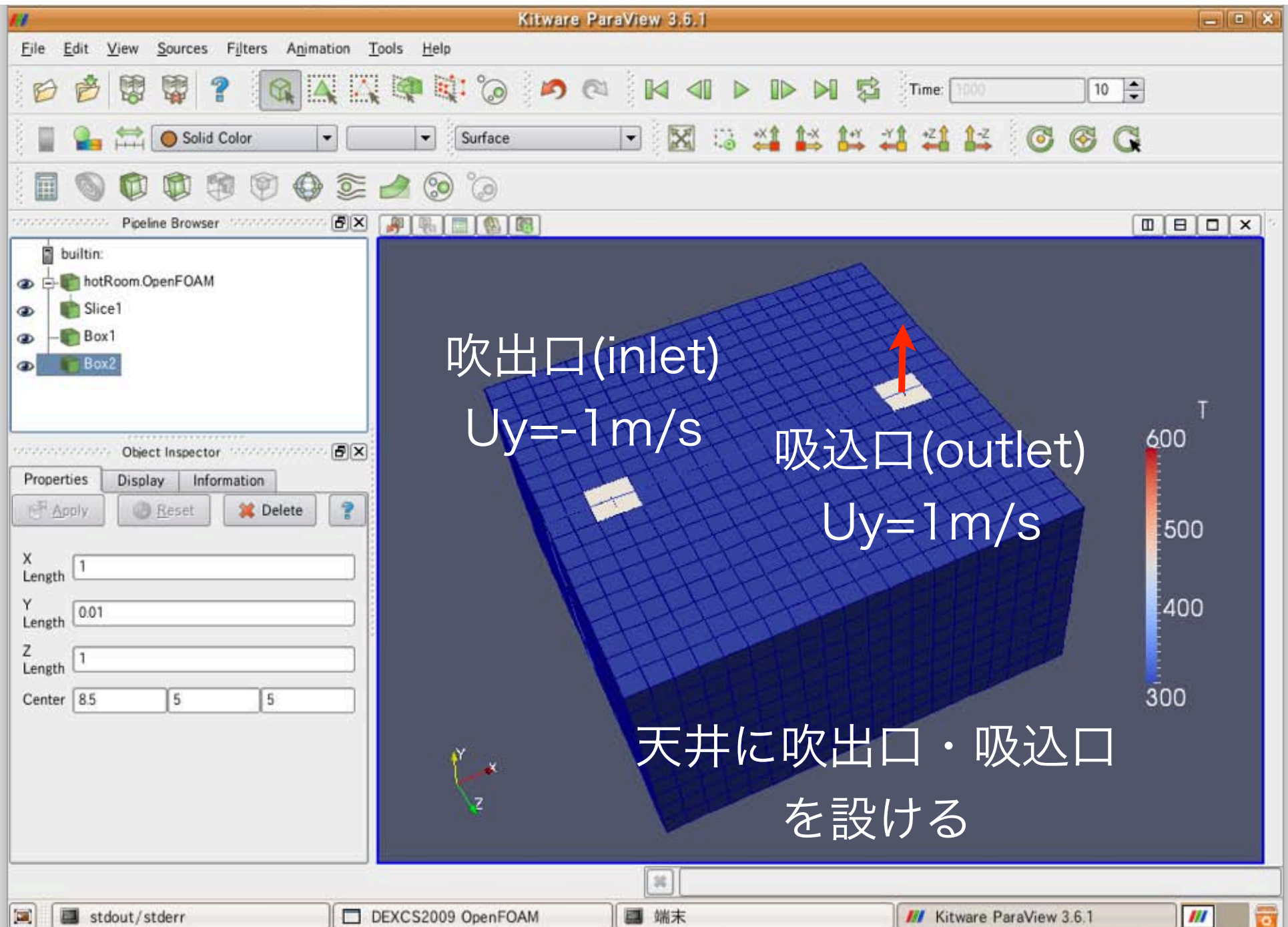
foamJob buoyantBoussinesqSimpleFoam ↵

tail -f log

↓ tailコマンドでlogを監視



吹出口・吸込口に対応する 新たなパッチ作成と初期値 の設定



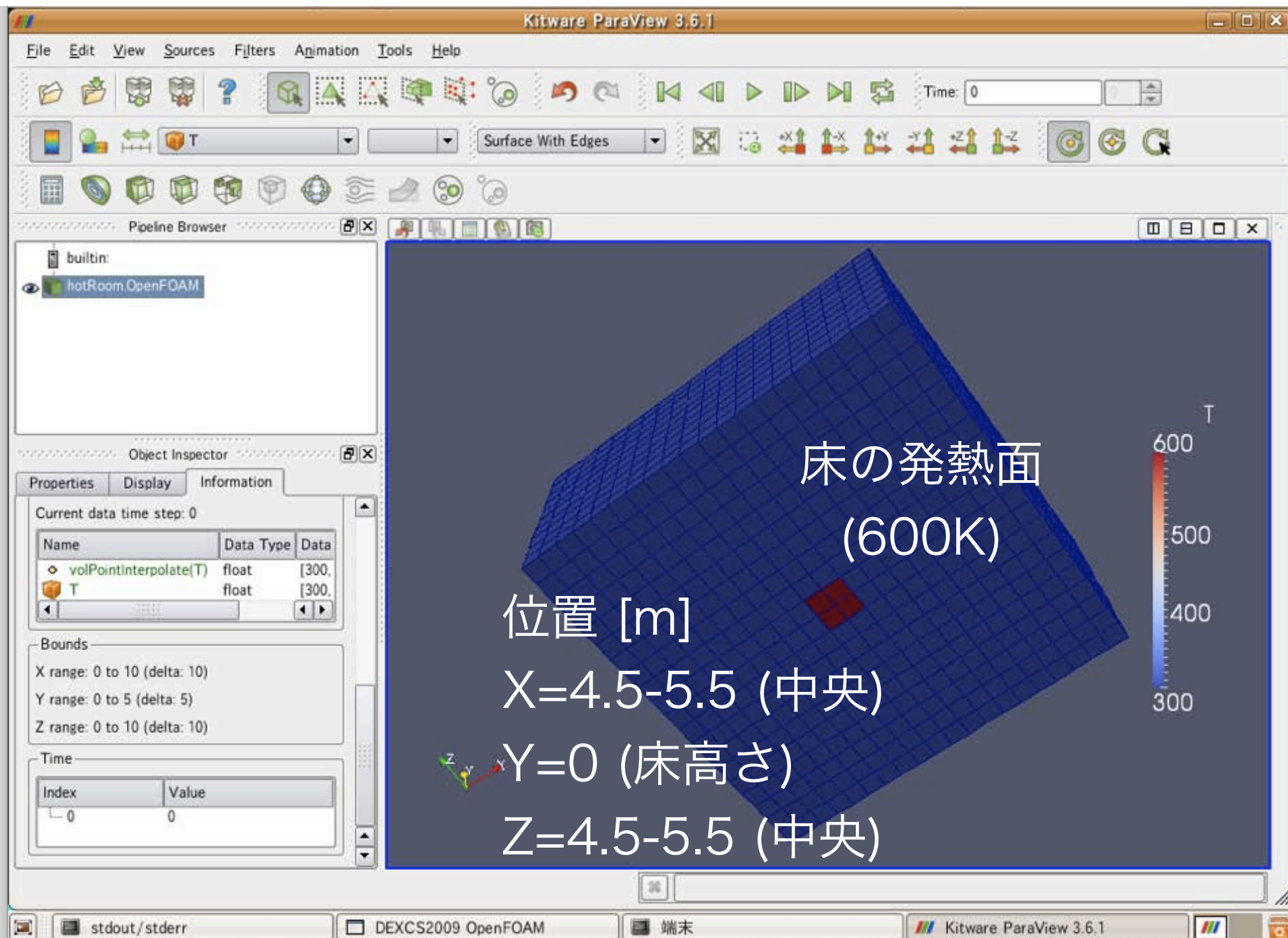
計算結果の消去

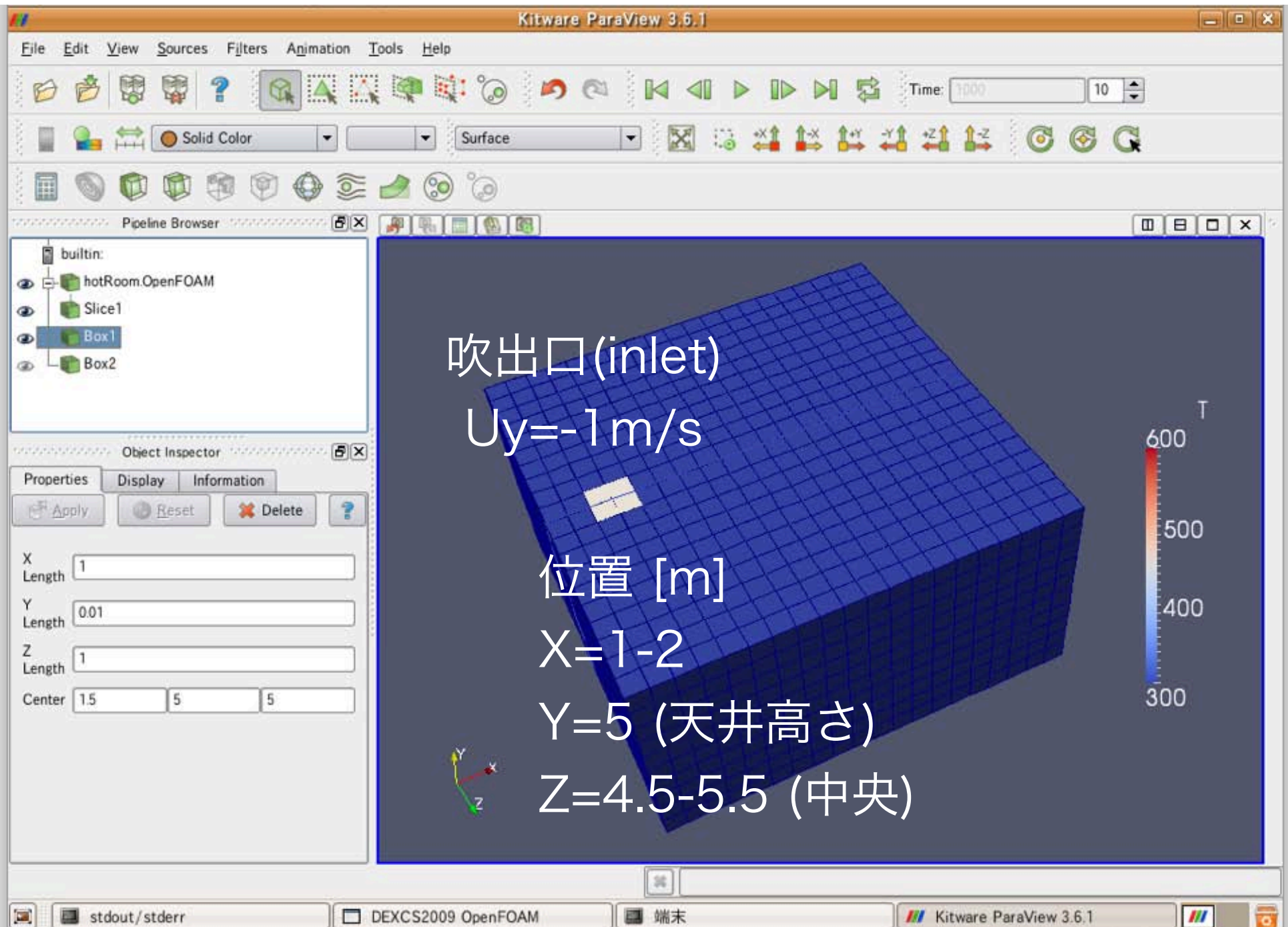
```
rm -r [1-9]* ↵
```

1. rmのrオプションは、ディレクトリごと消去
2. [1-9]は、1,2,...,9と展開される。[a-c]なら、a,b,c
3. [1-9]* は、0以外の計算結果のディレクトリ全てを表わす

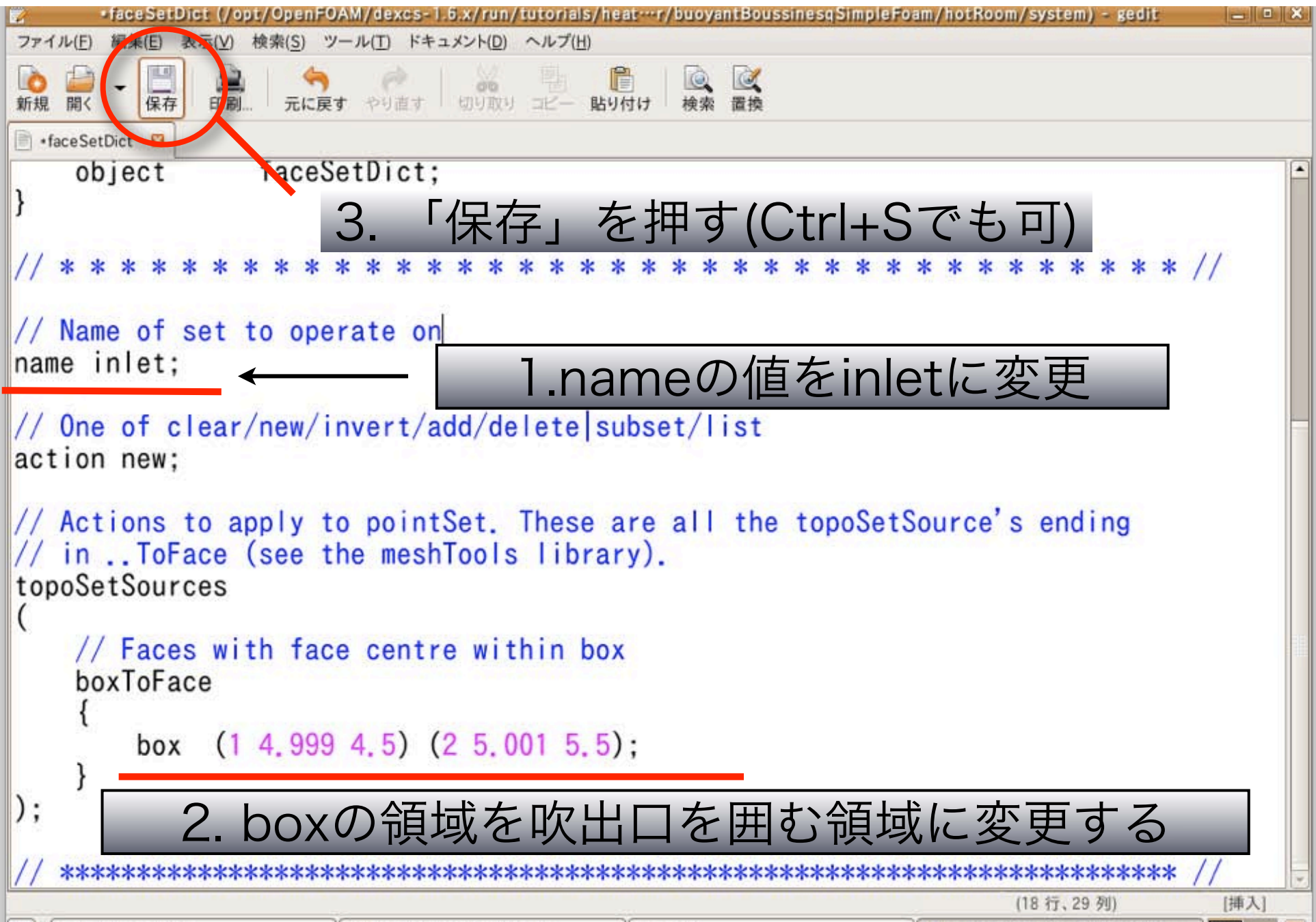
faceSetDictの修正

```
gedit system/faceSetDict & ↵
```





faceSetDictの修正



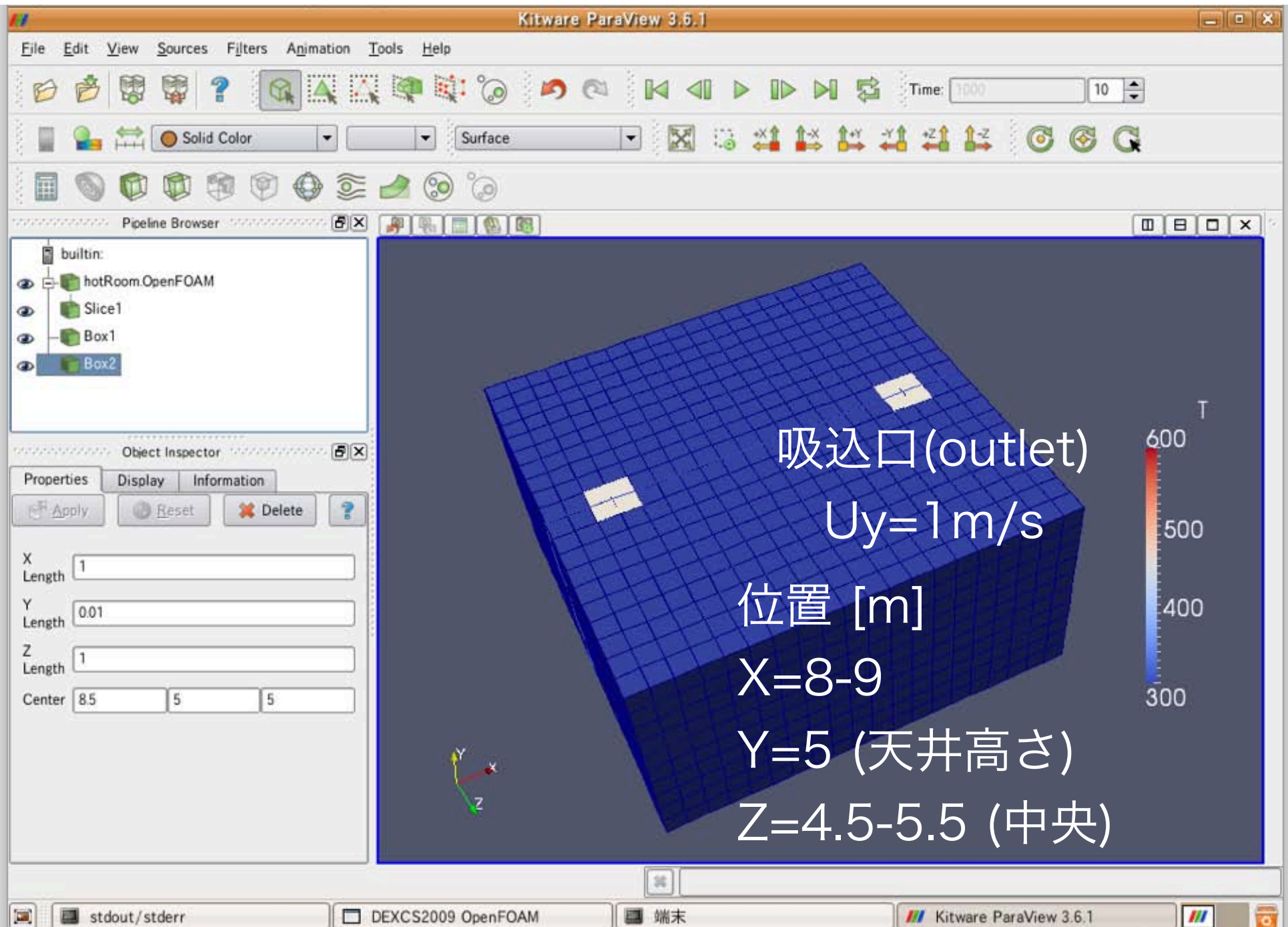
faceSetの実行

faceSet ↩

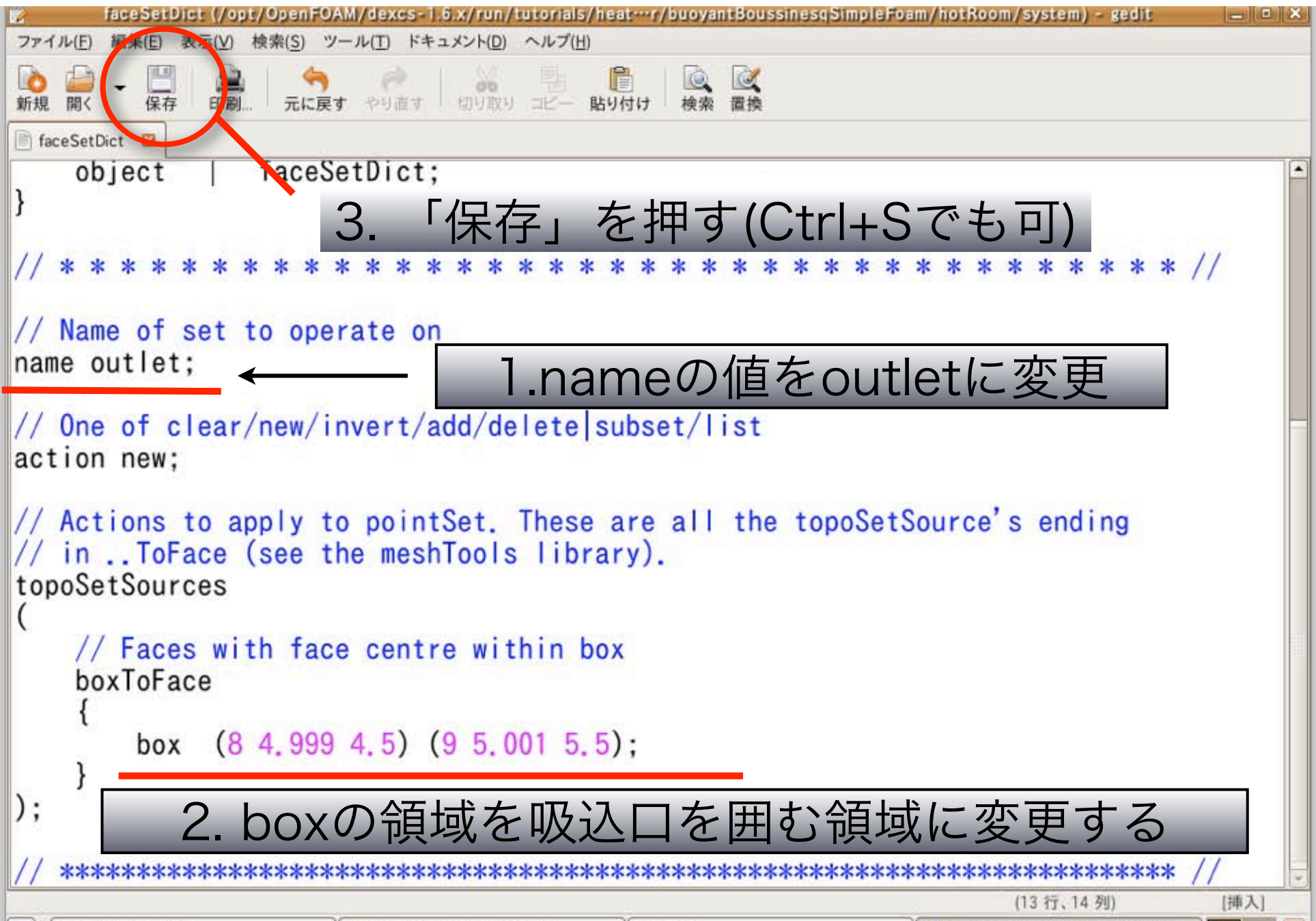
出力

```
Reading faceSetDict
Backing up inlet into inlet_old
Set:inlet Size:0 Action:new
    Adding faces with centre within box (1 4.999 4.5) (2
5.001 5.5)
Writing inlet (size 4) to "constant/polyMesh/sets/inlet"
```

4つの格子界面のfaceSetが、constant/polyMesh/sets/inletとして書き出された。



faceSetDictの修正



faceSetの実行

faceSet ↩

出力

```
Reading faceSetDict
```

```
Backing up outlet into outlet_old
```

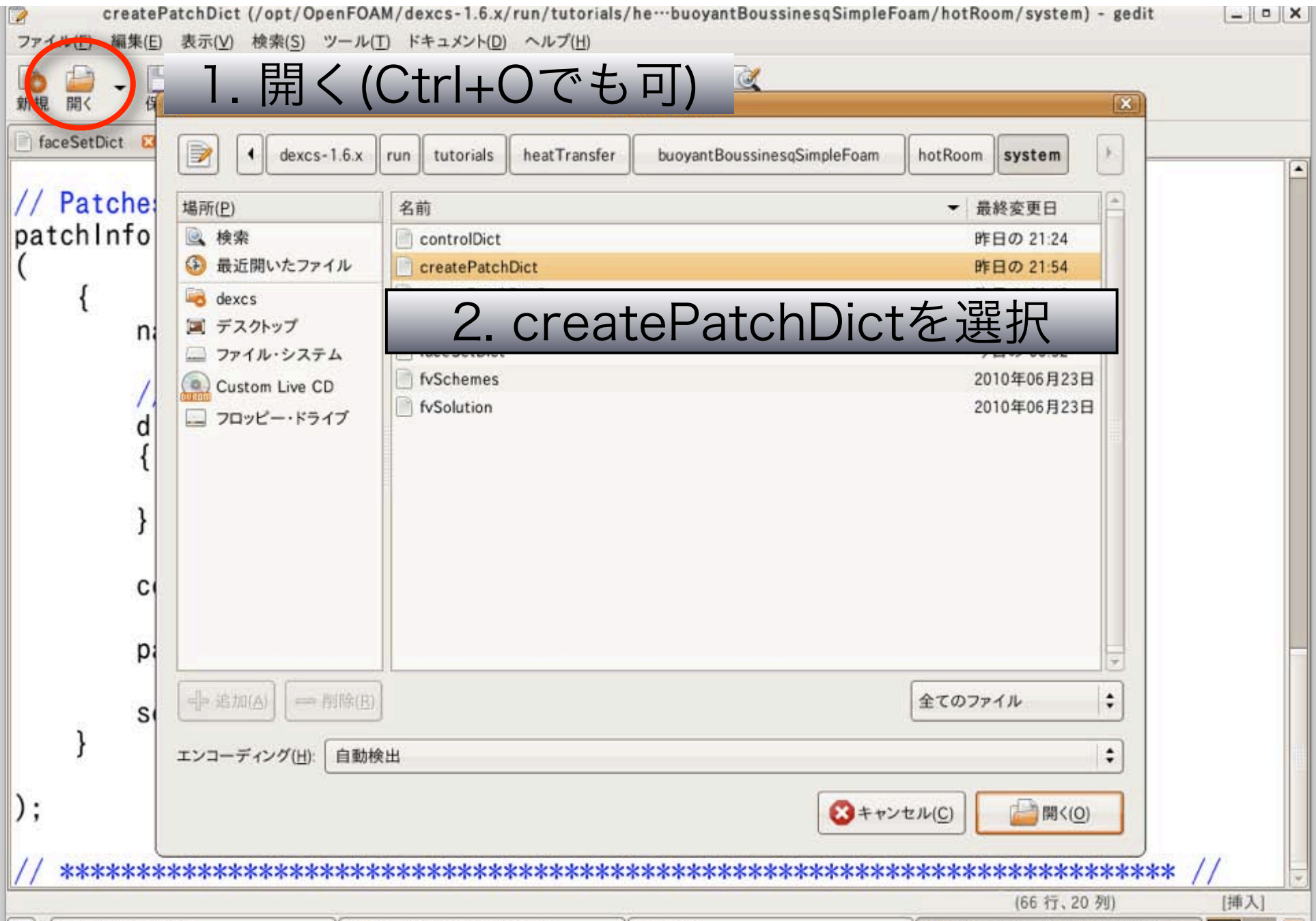
```
Set:outlet Size:0 Action:new
```

```
    Adding faces with centre within box (8 4.999 4.5) (9  
5.001 5.5)
```

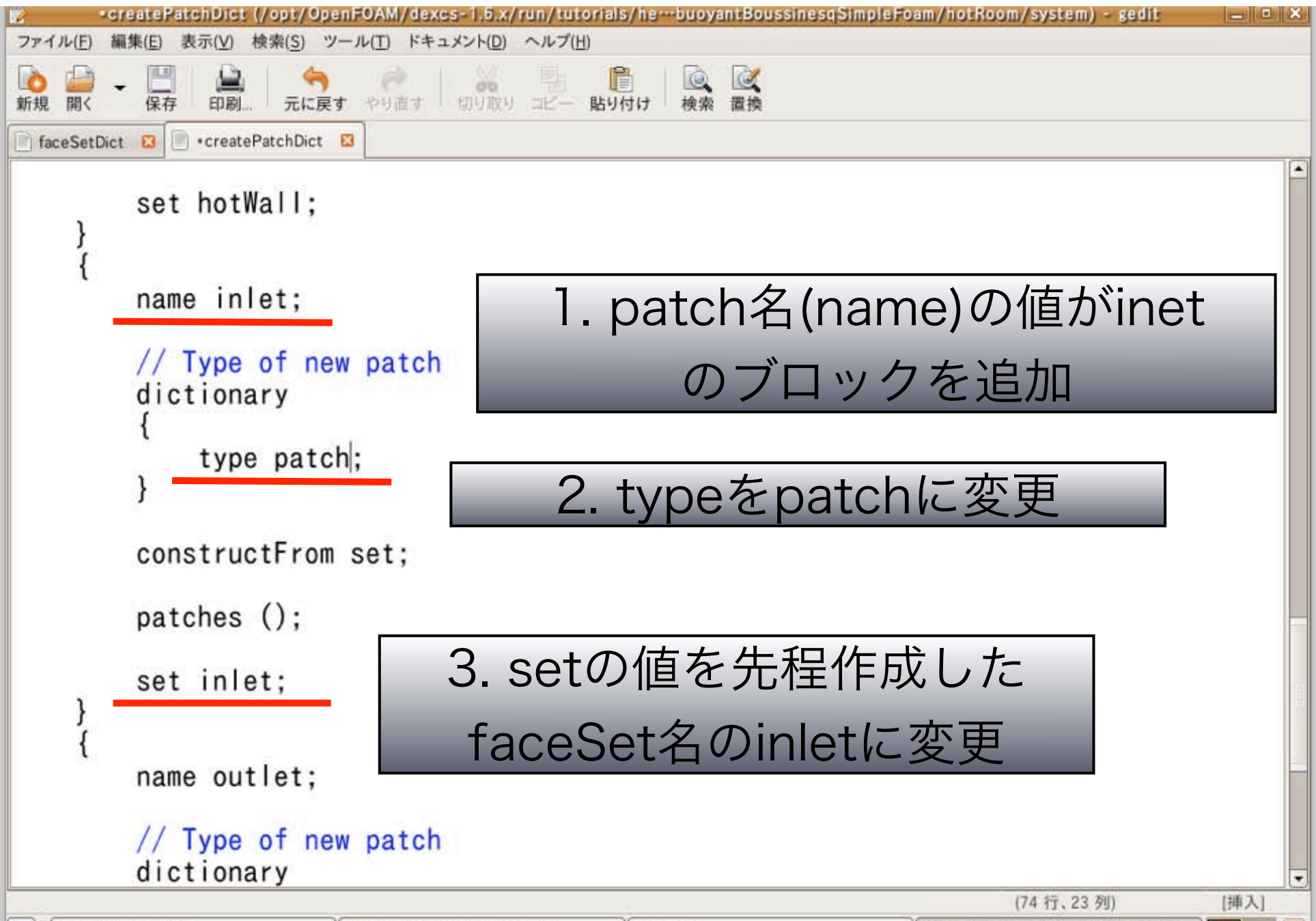
```
Writing outlet (size 4) to "constant/polyMesh/sets/outlet"
```

4つの格子界面のfaceSetが、constant/polyMesh/sets/outletとして書き出された。

createPatchDictの修正



createPatchDictの修正



createPatchDictの修正

```
patches ();
set inlet;
}
{
  name outlet;
  // Type of new patch
  dictionary
  {
    type patch;
  }
  constructFrom set;
  patches ();
  set outlet;
}
);
// ***** //
```

3. 「保存」を押す(Ctrl+Sでも可)

1. patch名(name)の値がoutlet
のブロックを追加

2. typeをpatchに変更

3. setの値を先程作成した
faceSet名のoutletに変更

(88 行、10 列) [挿入]

createPatchの実行

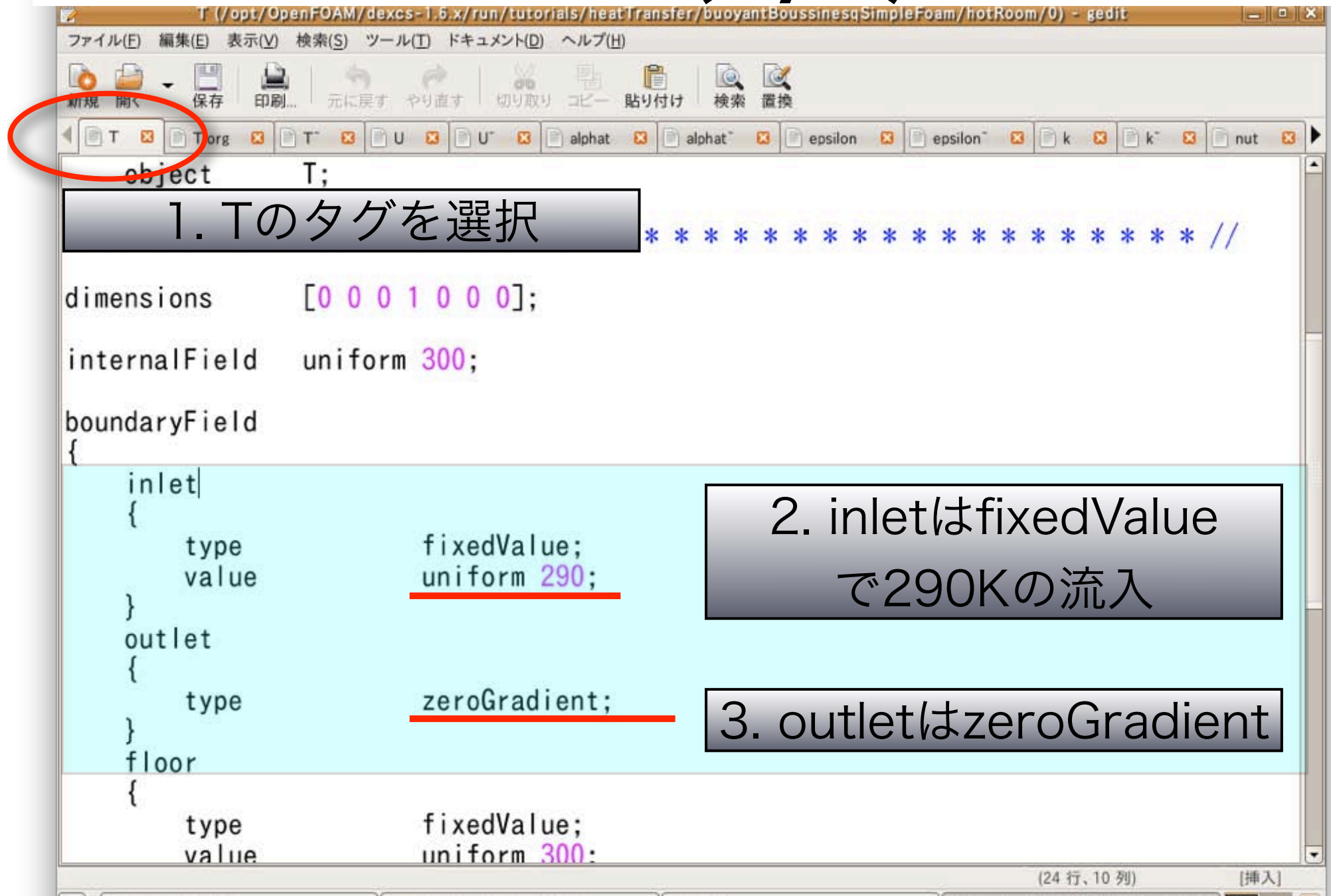
```
createPatch -overwrite↵
```

inletやoutletのfaceSetからなる新しいpatchである
inletやoutletがconstant/polyMeshに上書きで出力

各物理量の初期ファイルの修正

```
gedit 0/* &↵
```


初期ファイル(0/T)の修正



The screenshot shows a gedit editor window with the file path `T (/opt/OpenFOAM/dexcs-1.6.x/run/tutorials/heatTransfer/buoyantBoussinesqSimpleFoam/hotRoom/0) - gedit`. The file content is as follows:

```
object T;

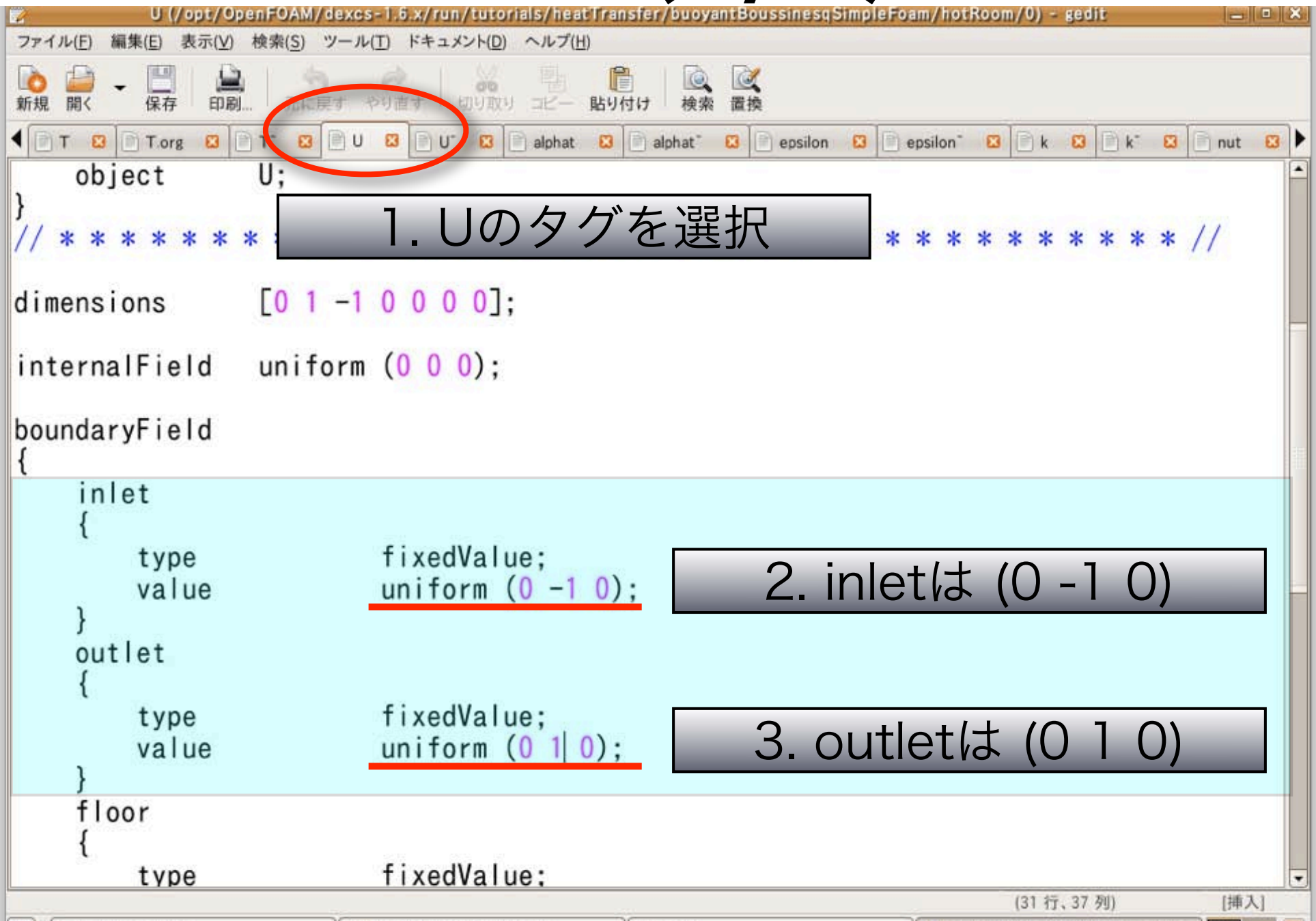
dimensions      [0 0 0 1 0 0 0];
internalField    uniform 300;

boundaryField
{
    inlet
    {
        type      fixedValue;
        value      uniform 290;
    }
    outlet
    {
        type      zeroGradient;
    }
    floor
    {
        type      fixedValue;
        value      uniform 300;
    }
}
```

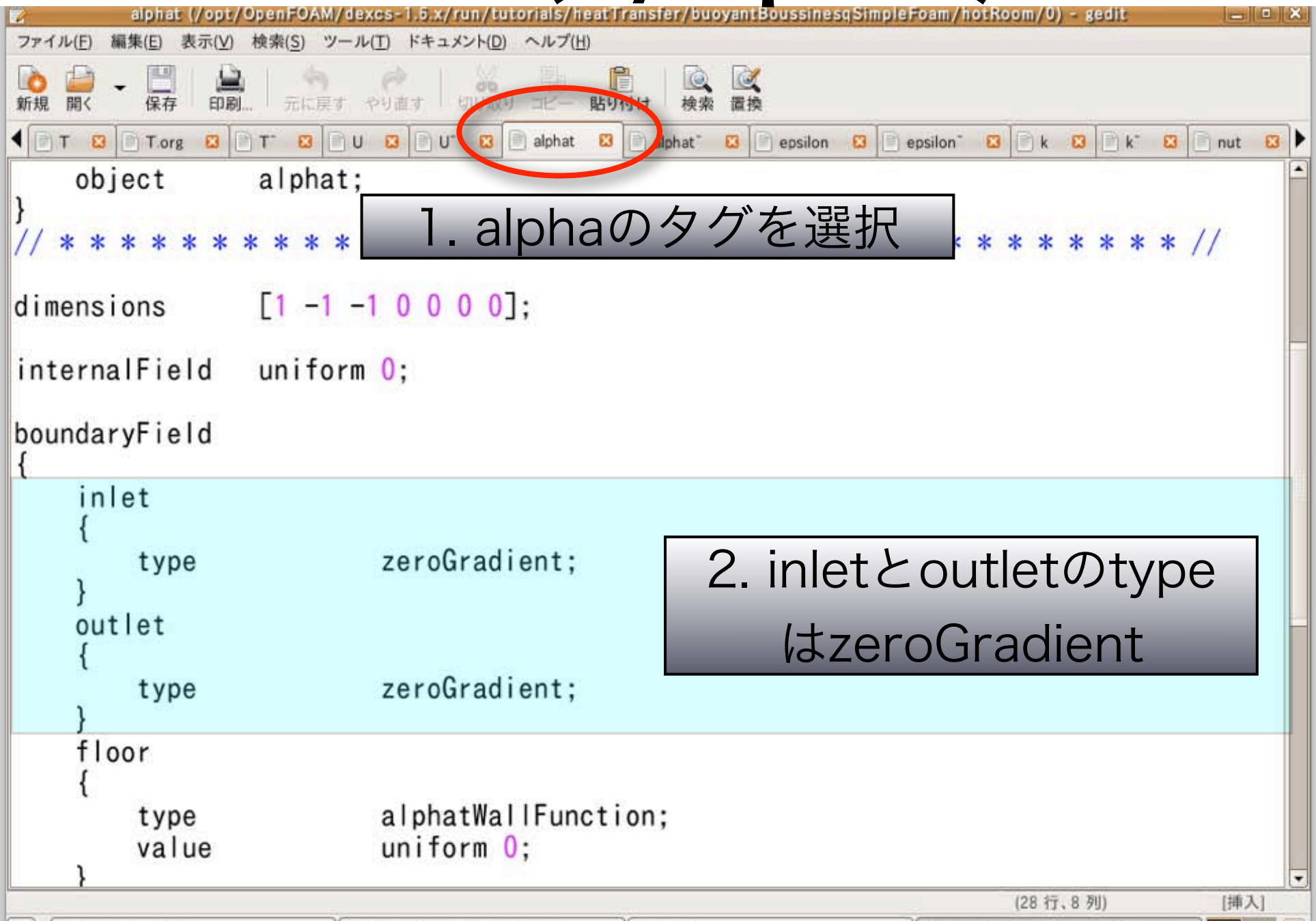
Annotations and instructions:

- 1. Tのタグを選択
- 2. inletはfixedValueで290Kの流入
- 3. outletはzeroGradient

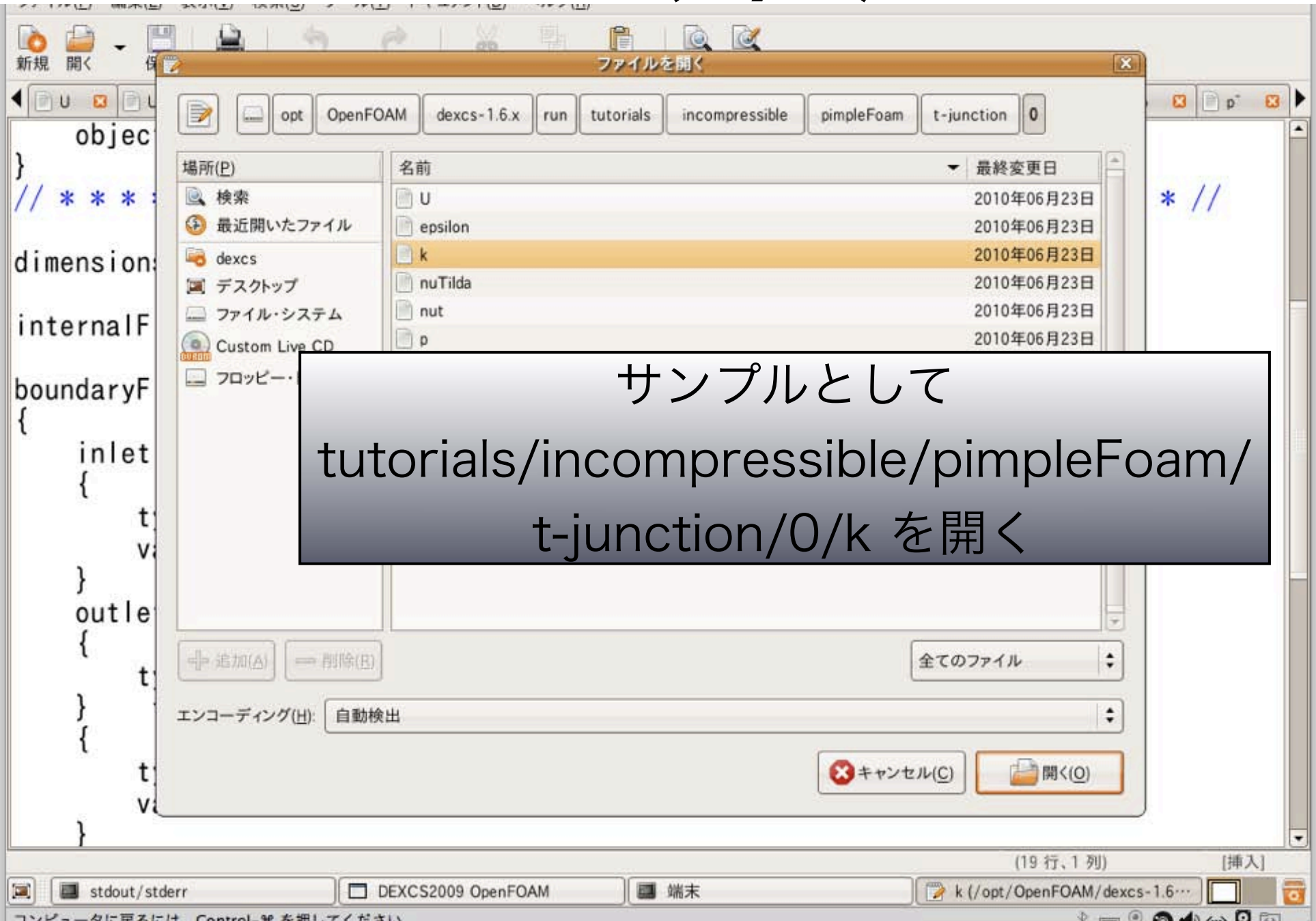
初期ファイル(0/U)の修正



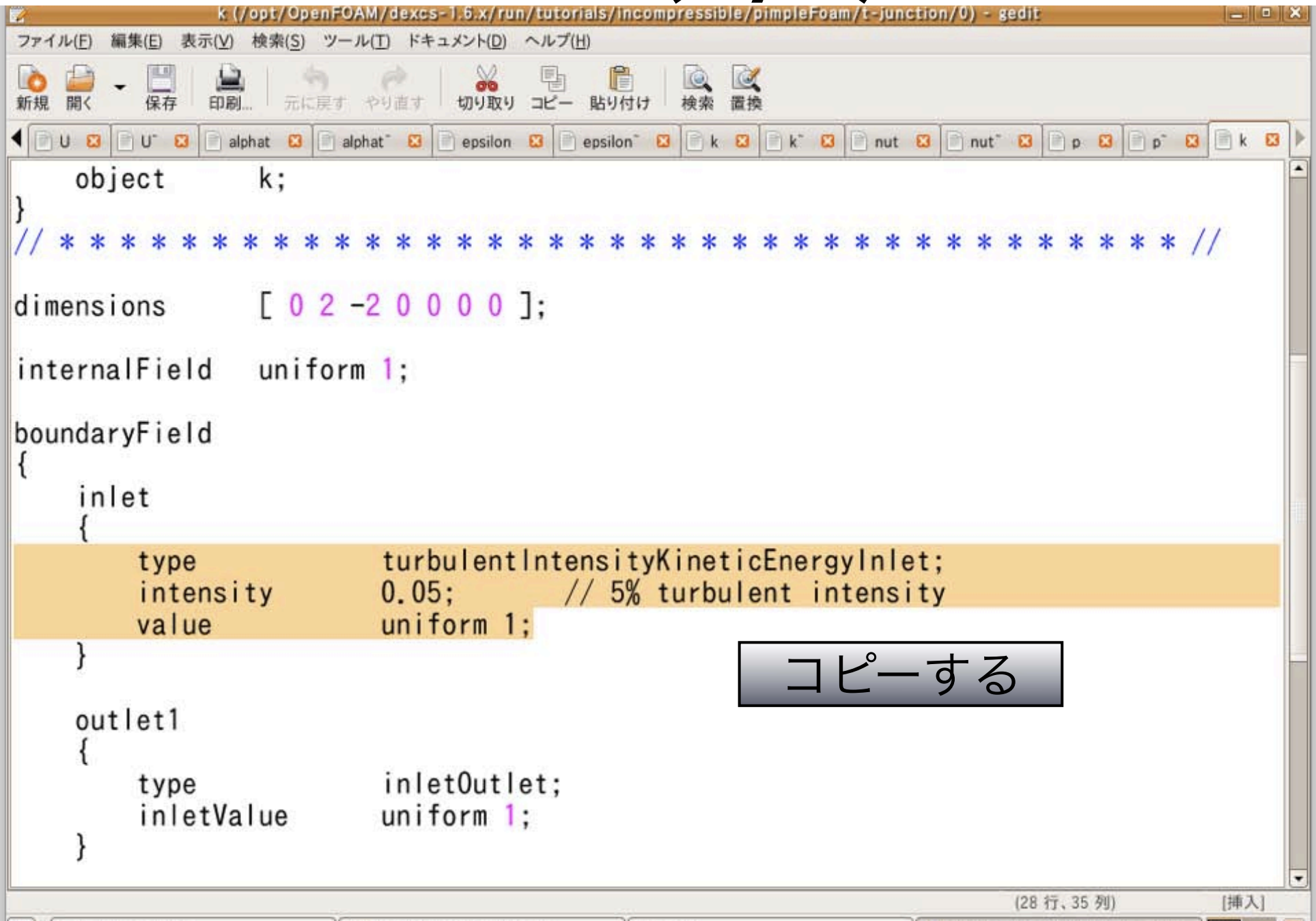
初期ファイル(0/alphat)の修正



初期ファイル(0/k)の修正



初期ファイル(0/k)の修正



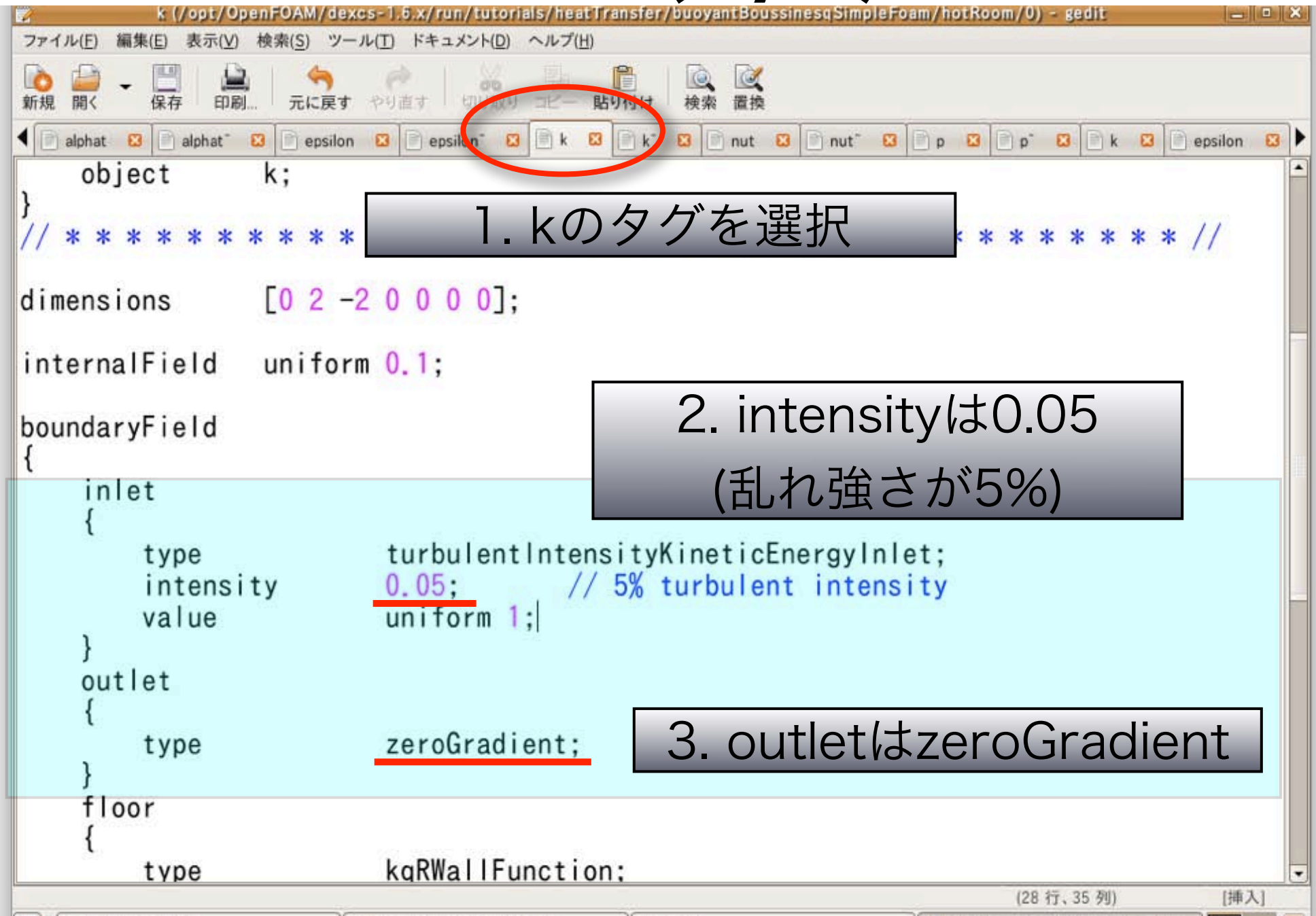
```
k (/opt/OpenFOAM/dexcs-1.6.x/run/tutorials/incompressible/pimpleFoam/t-junction/0) - gedit
ファイル(E) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
新規 開く 保存 印刷... 元に戻す やり直す 切り取り コピー 貼り付け 検索 置換
U U~ alphas alphas~ epsilon epsilon~ k k~ nut nut~ p p~ k
object      k;
}
// ****

dimensions  [ 0 2 -2 0 0 0 0 ];
internalField  uniform 1;
boundaryField
{
    inlet
    {
        type          turbulentIntensityKineticEnergyInlet;
        intensity      0.05;          // 5% turbulent intensity
        value           uniform 1;
    }
    outlet1
    {
        type          inletOutlet;
        inletValue     uniform 1;
    }
}
```

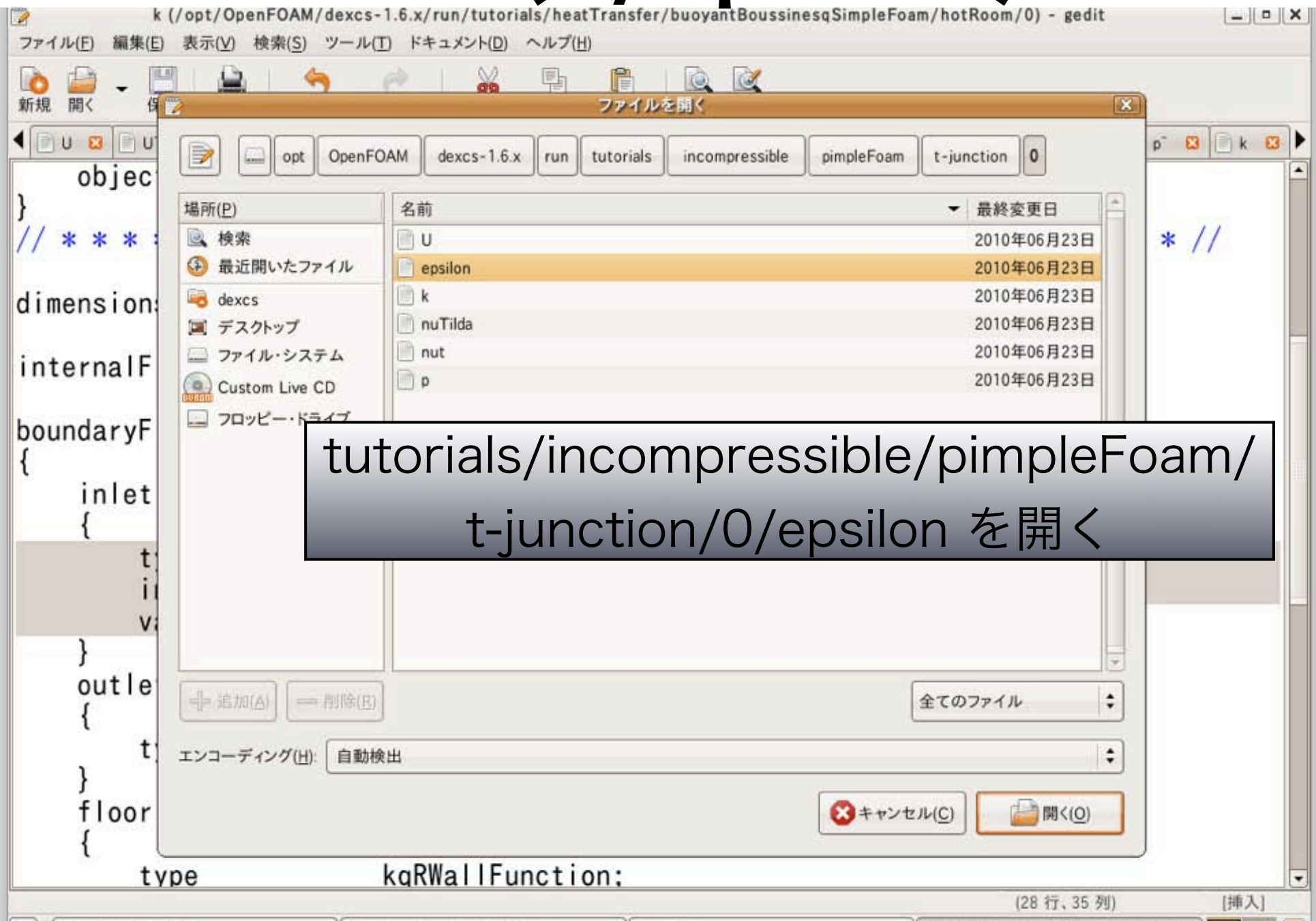
コピーする

(28 行, 35 列) [挿入]

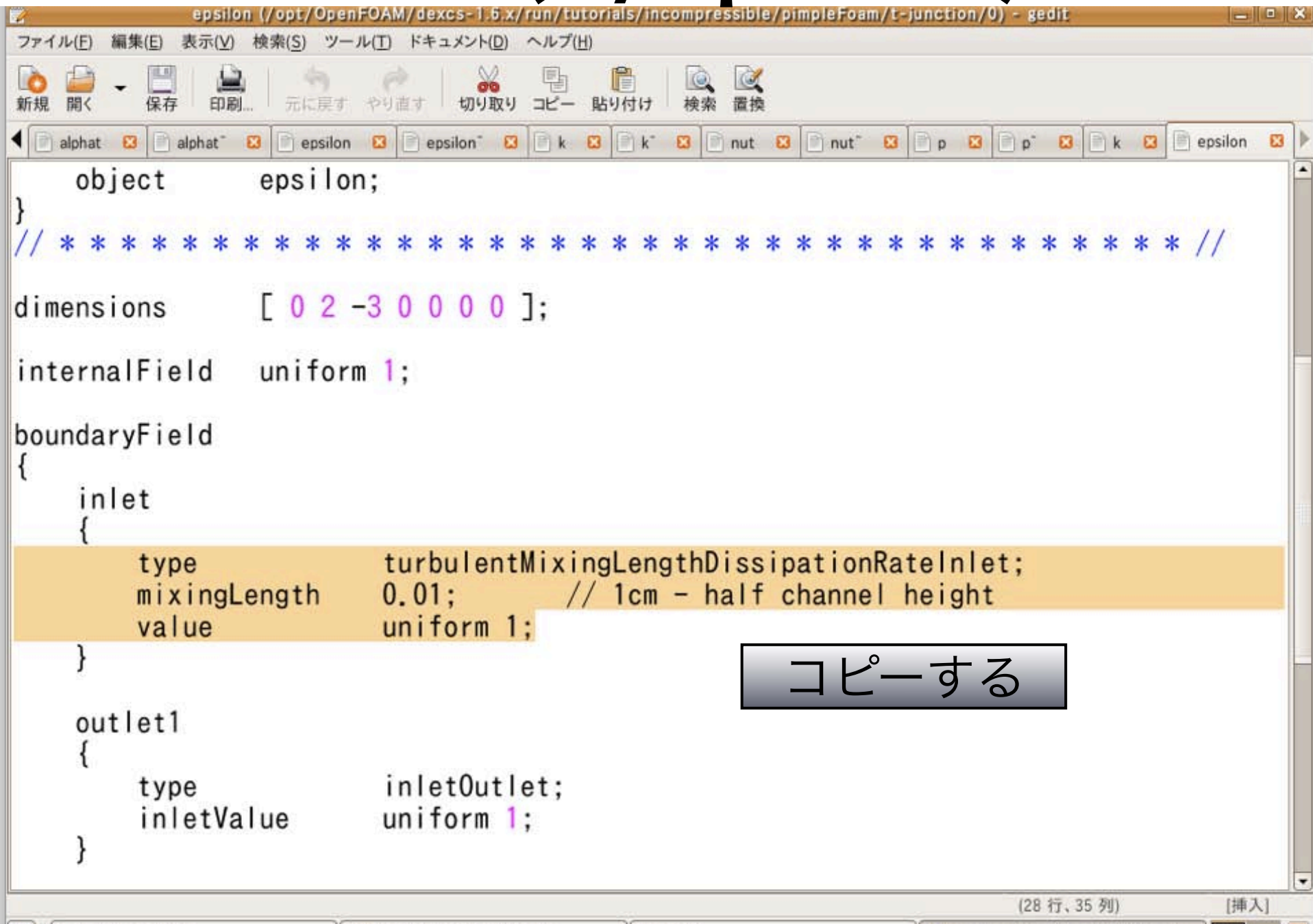
初期ファイル(0/k)の修正



初期ファイル(0/epsilon)の修正



初期ファイル(0/epsilon)の修正



```
object      epsilon;
}
// *****

dimensions      [ 0 2 -3 0 0 0 0 ];
internalField    uniform 1;
boundaryField
{
    inlet
    {
        type          turbulentMixingLengthDissipationRateInlet;
        mixingLength    0.01;          // 1cm - half channel height
        value            uniform 1;
    }

    outlet1
    {
        type          inletOutlet;
        inletValue      uniform 1;
    }
}
```

コピーする

初期ファイル(0/epsilon)の修正

epsilon (/opt/OpenFOAM/dexcs-1.6.x/run/tutorials/heatTransfer/buoyantBoussinesqSimpleFoam/hotRoom/0) - gedit

ファイル(E) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

新規 開く 保存 印刷 元に戻す やり直す 切り取り コピー 貼り付け 検索 置換

alphat alphat epsilon epsilon~ k k~ nut nut~ p p~ k epsilon

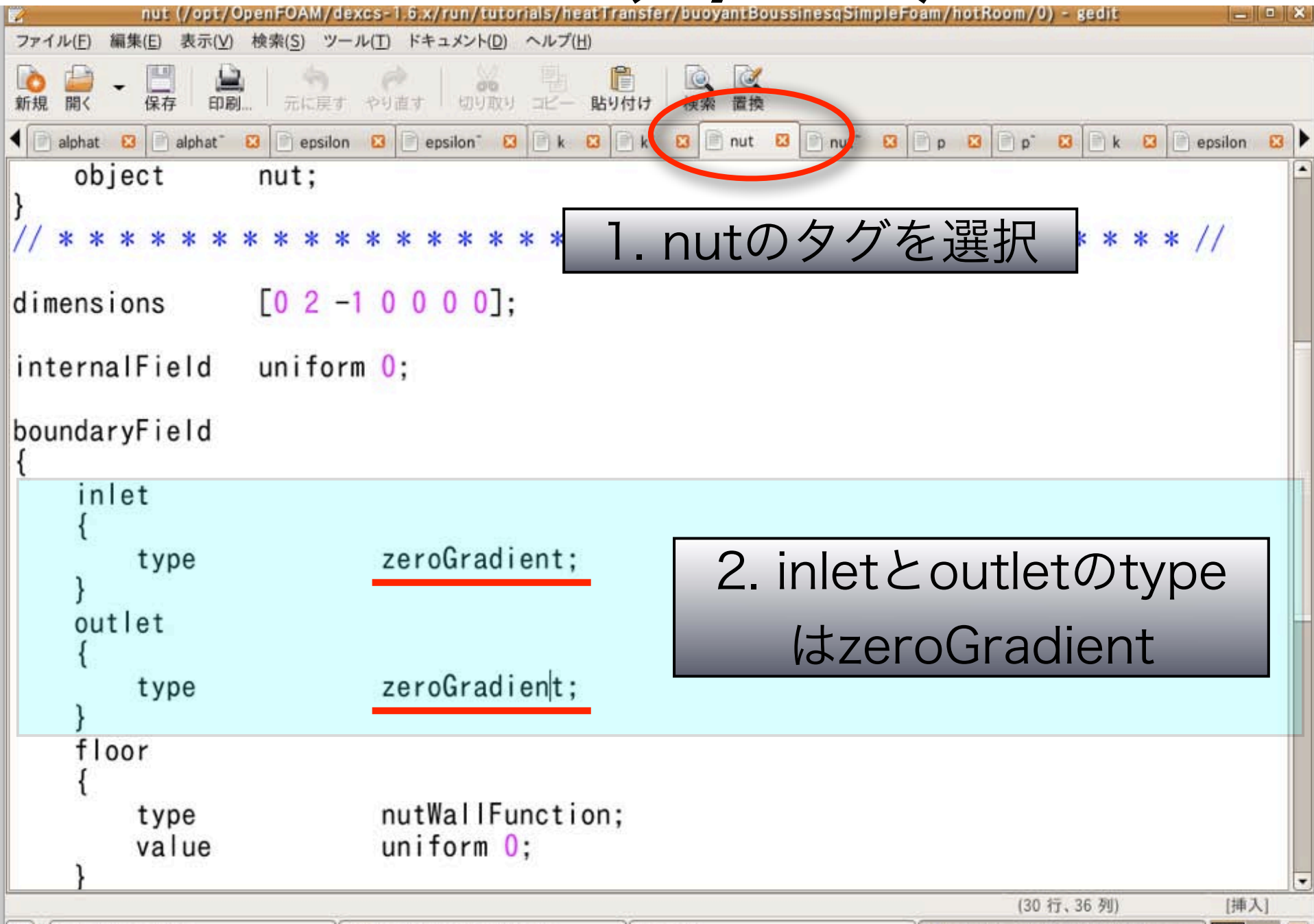
```
object
{
  // ***** 1. epsilonのタグを選択 ***** //
  dimensions      [0 2 -3 0 0 0 0];
  internalField    uniform 0.01;
  boundaryField
  {
    inlet
    {
      type          turbulentMixingLengthDissipationRateInlet;
      mixingLength  0.1; // 10cm
      value         uniform 1;
    }
    outlet
    {
      type          zeroGradient;
    }
    floor
    {
      type          epsilonWallFunction;
    }
  }
}
```

2. mixingLengthは0.1
(乱流渦長さスケールを
吹出口幅の1/10)

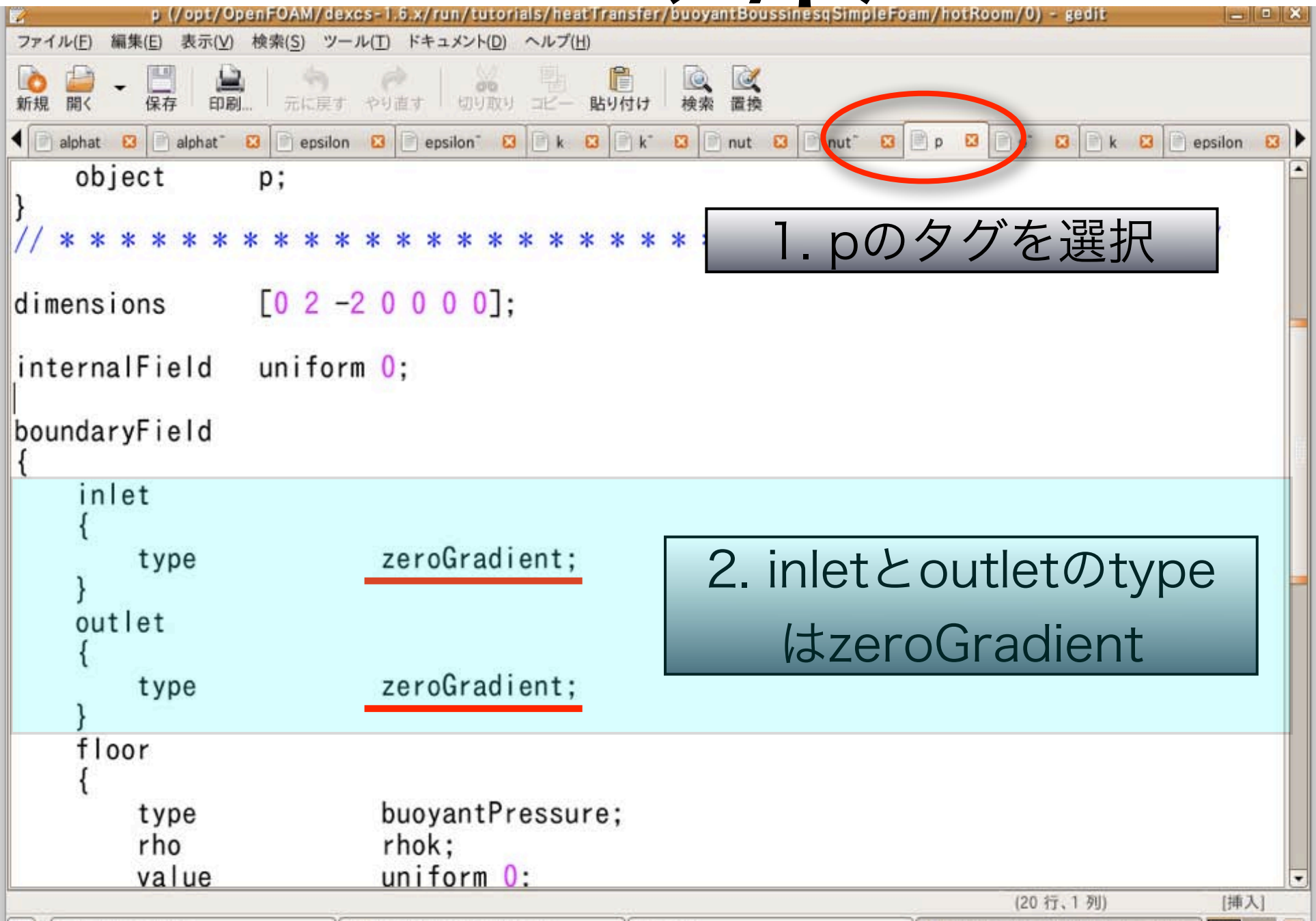
3. outletはzeroGradient

ファイル "/opt/OpenFOAM/dexcs-1.6.x/run/tutorials/heatTransfer/buoyantBoussinesqSimpleFoam/hotRoom/0/epsilon" を保存してい... (27 行, 34 列) [挿入]

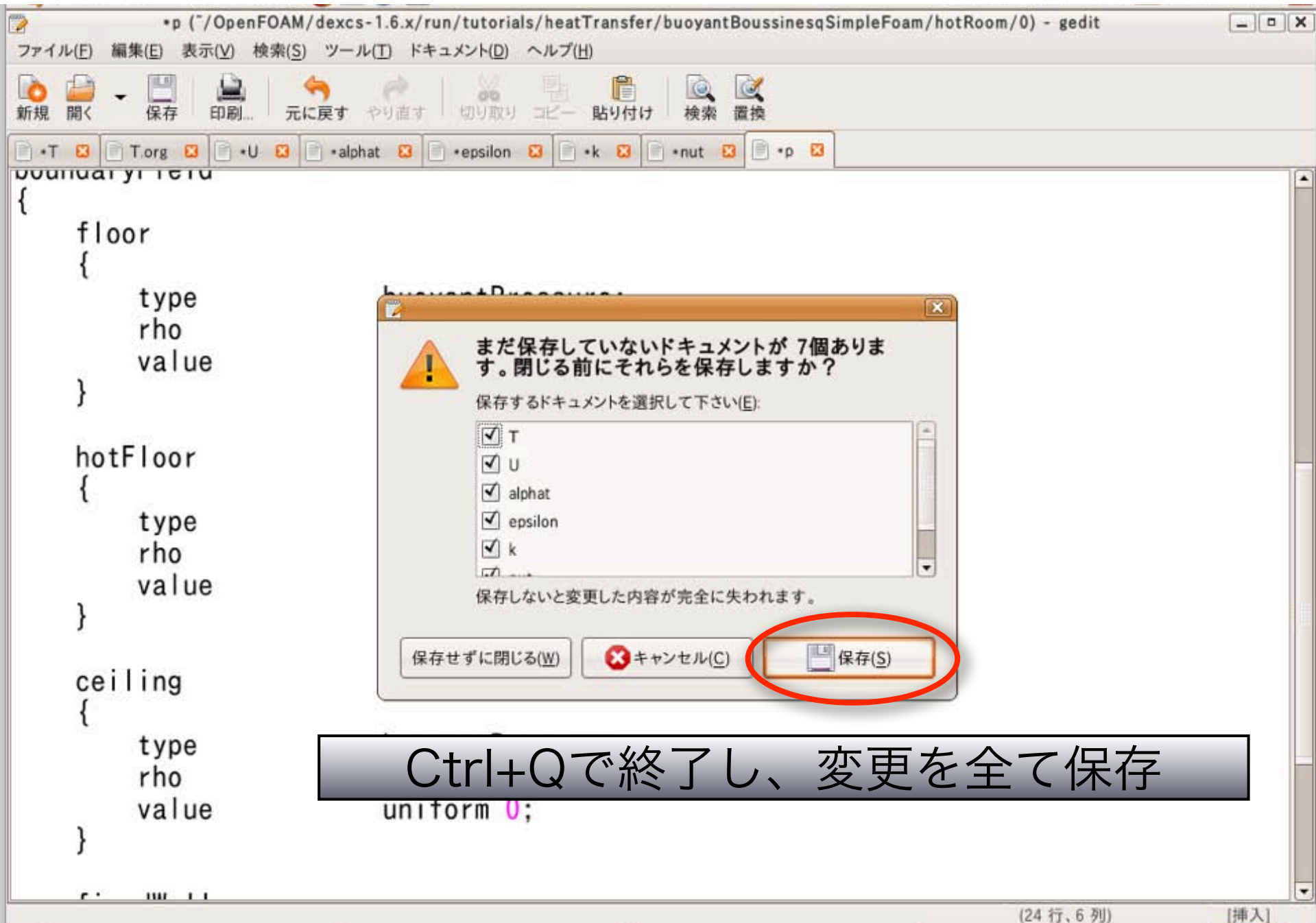
初期ファイル(0/nut)の修正



初期ファイル(0/p)の修正



その他の初期ファイルの修正



計算実行

```
buoyantBoussinesqSimpleFoam ↵
```

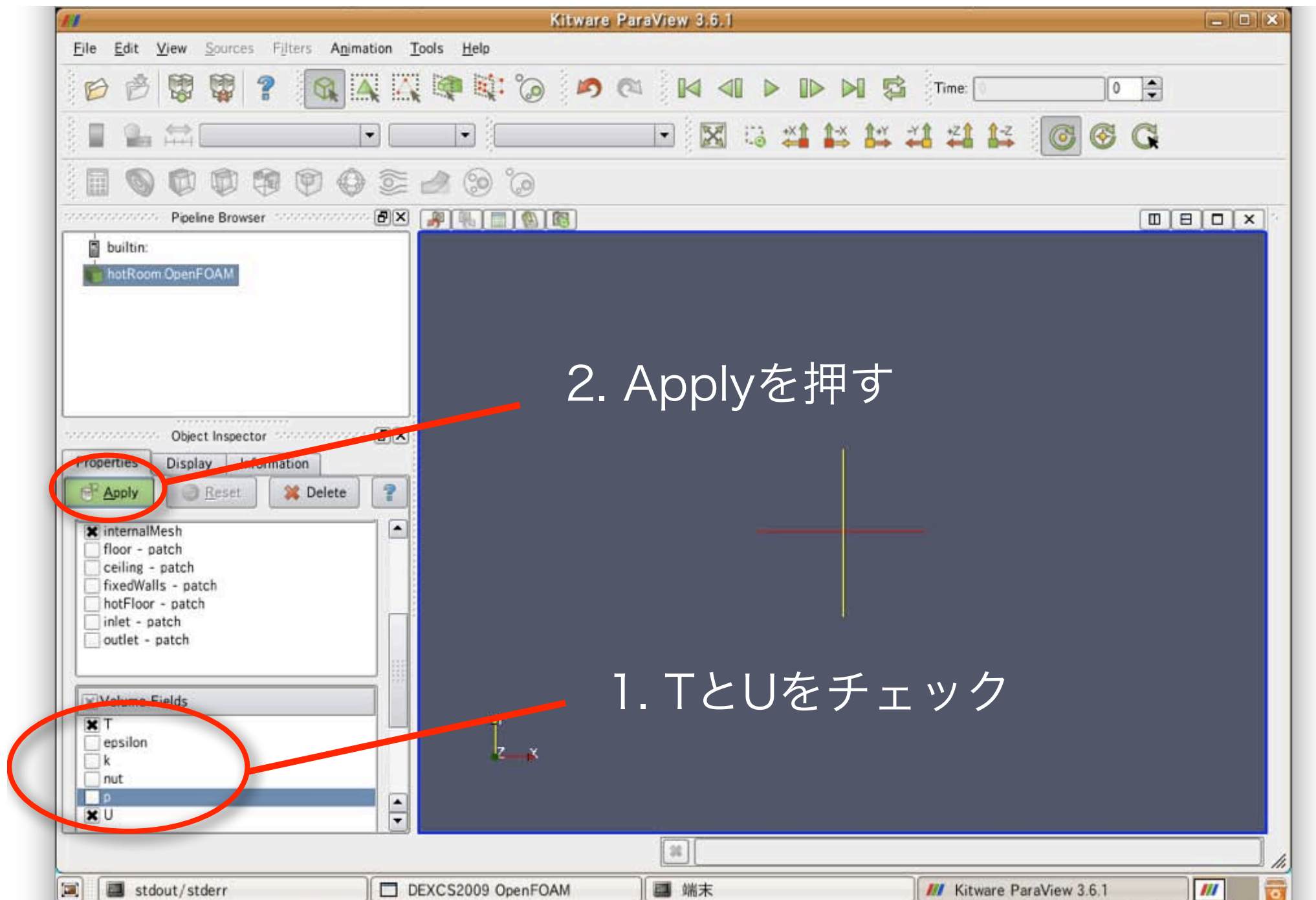
または

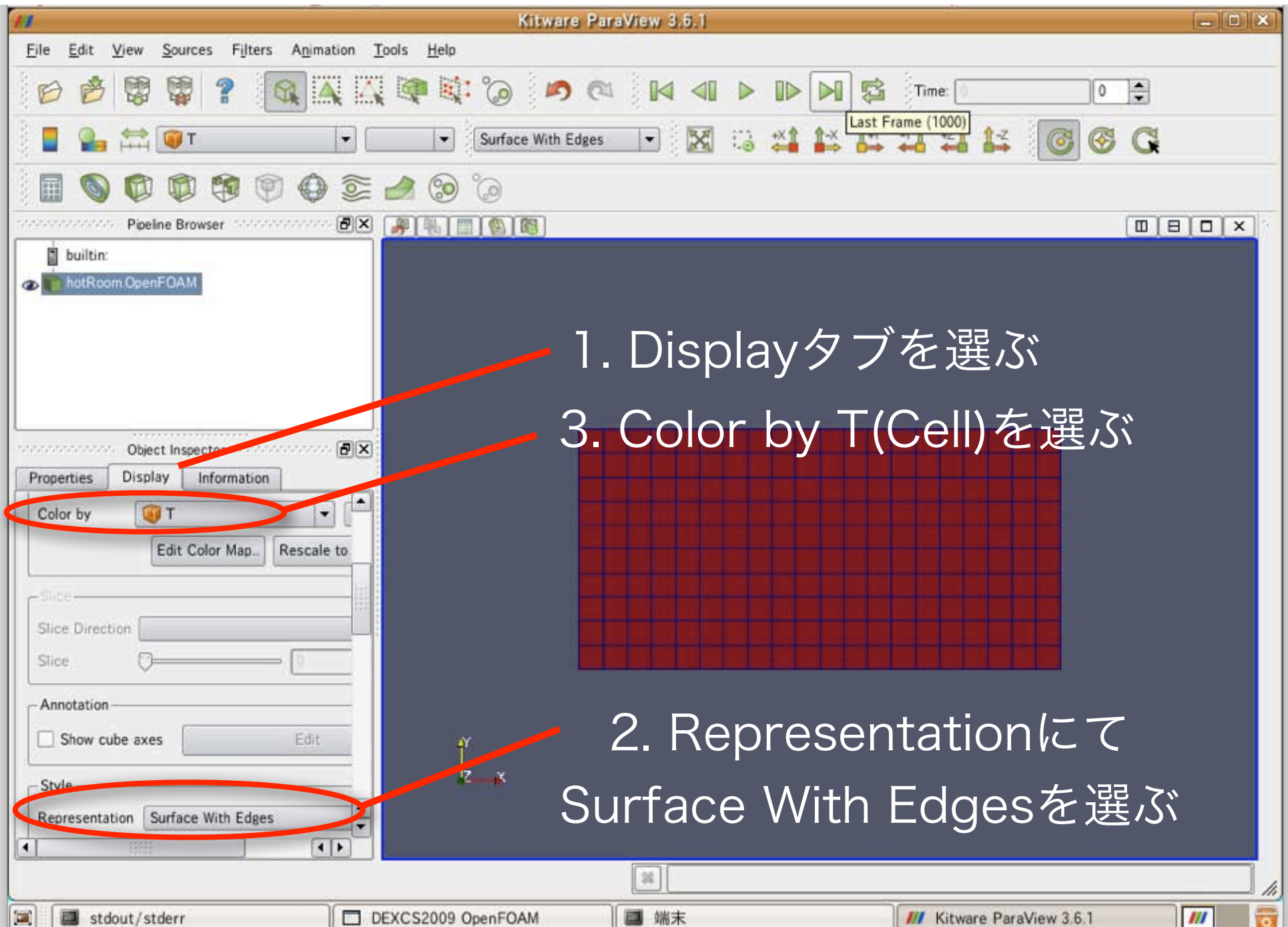
```
foamJob buoyantBoussinesqSimpleFoam ↵
```

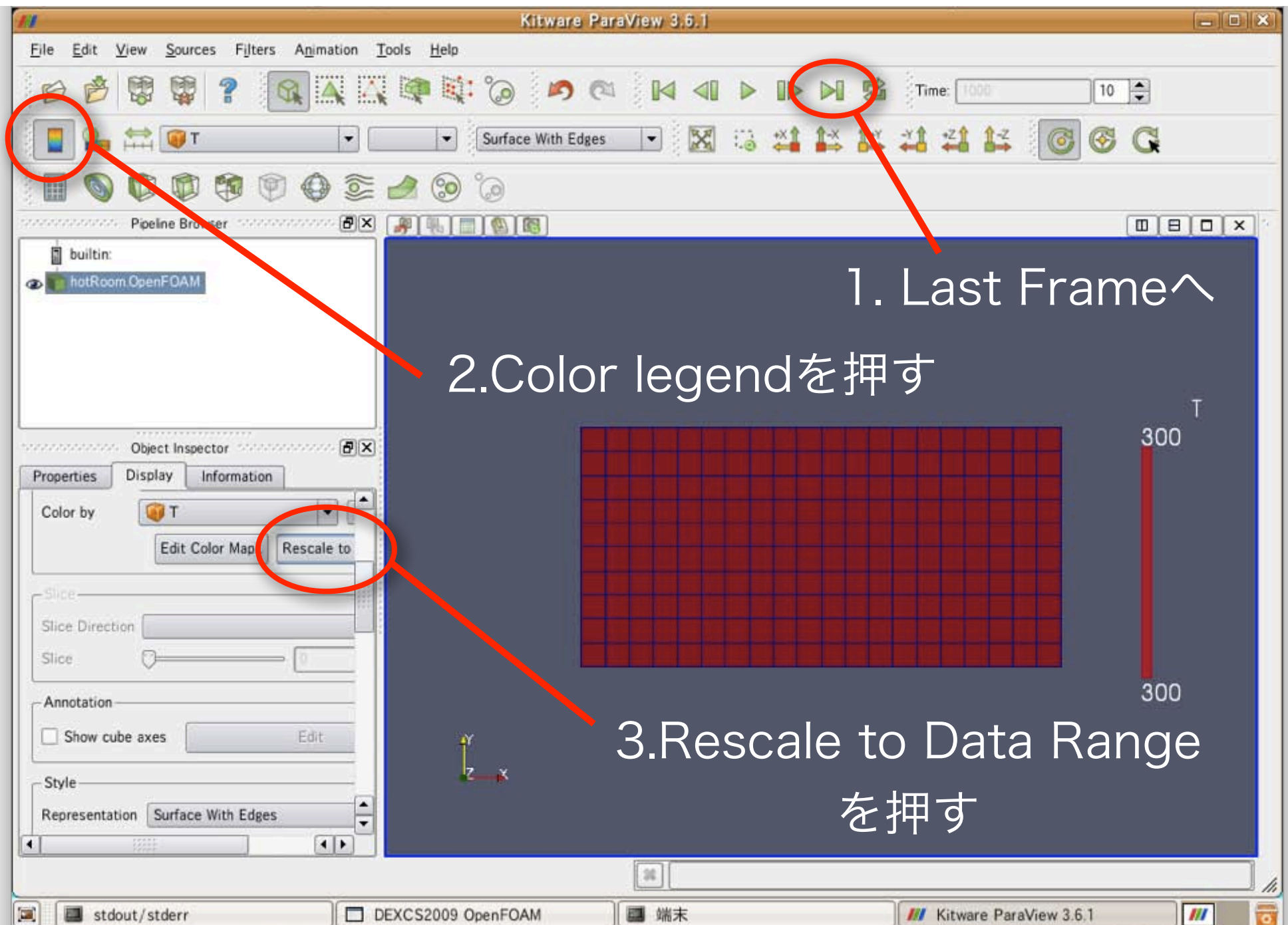
```
tail -f log
```

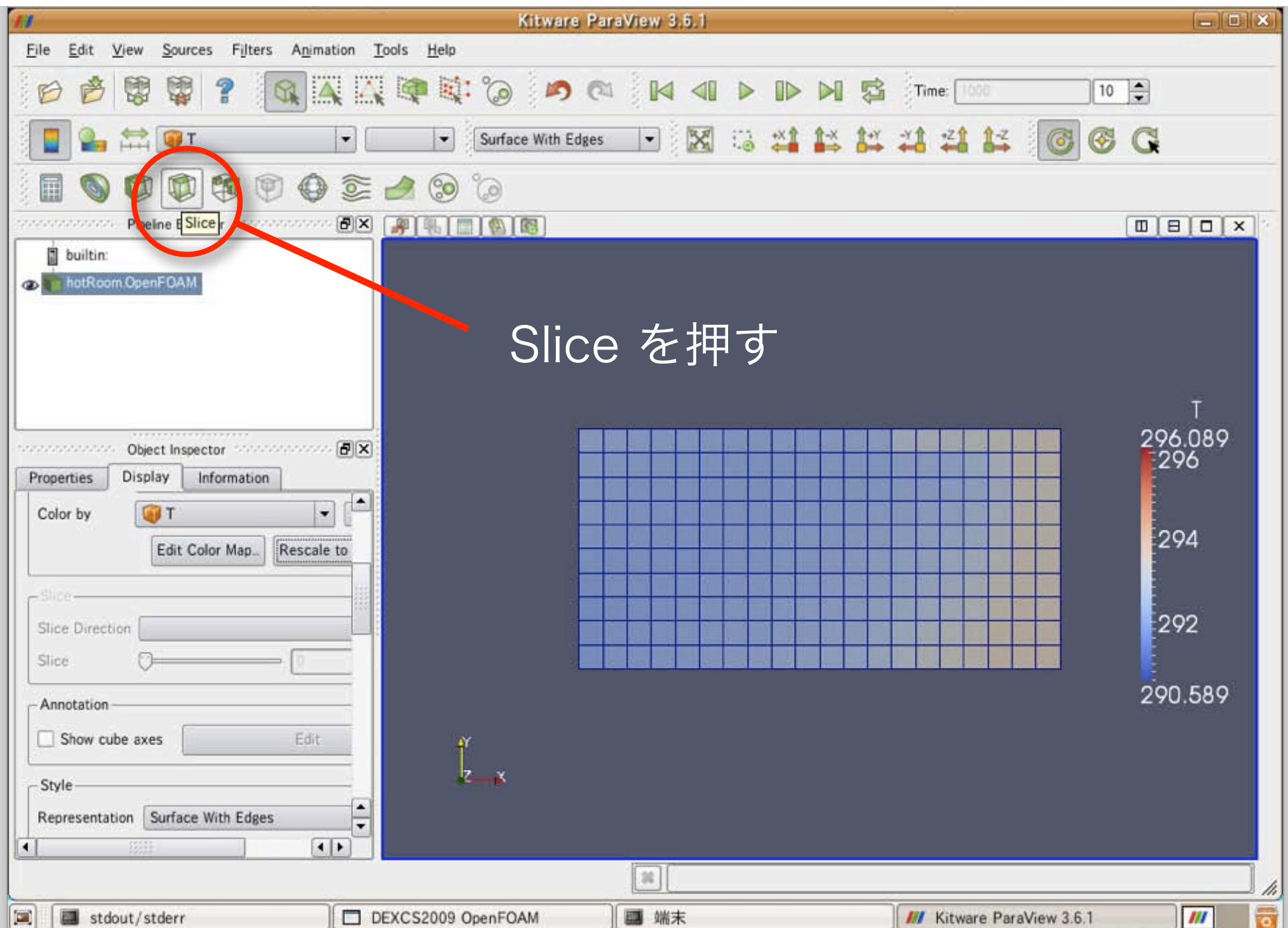
結果の可視化

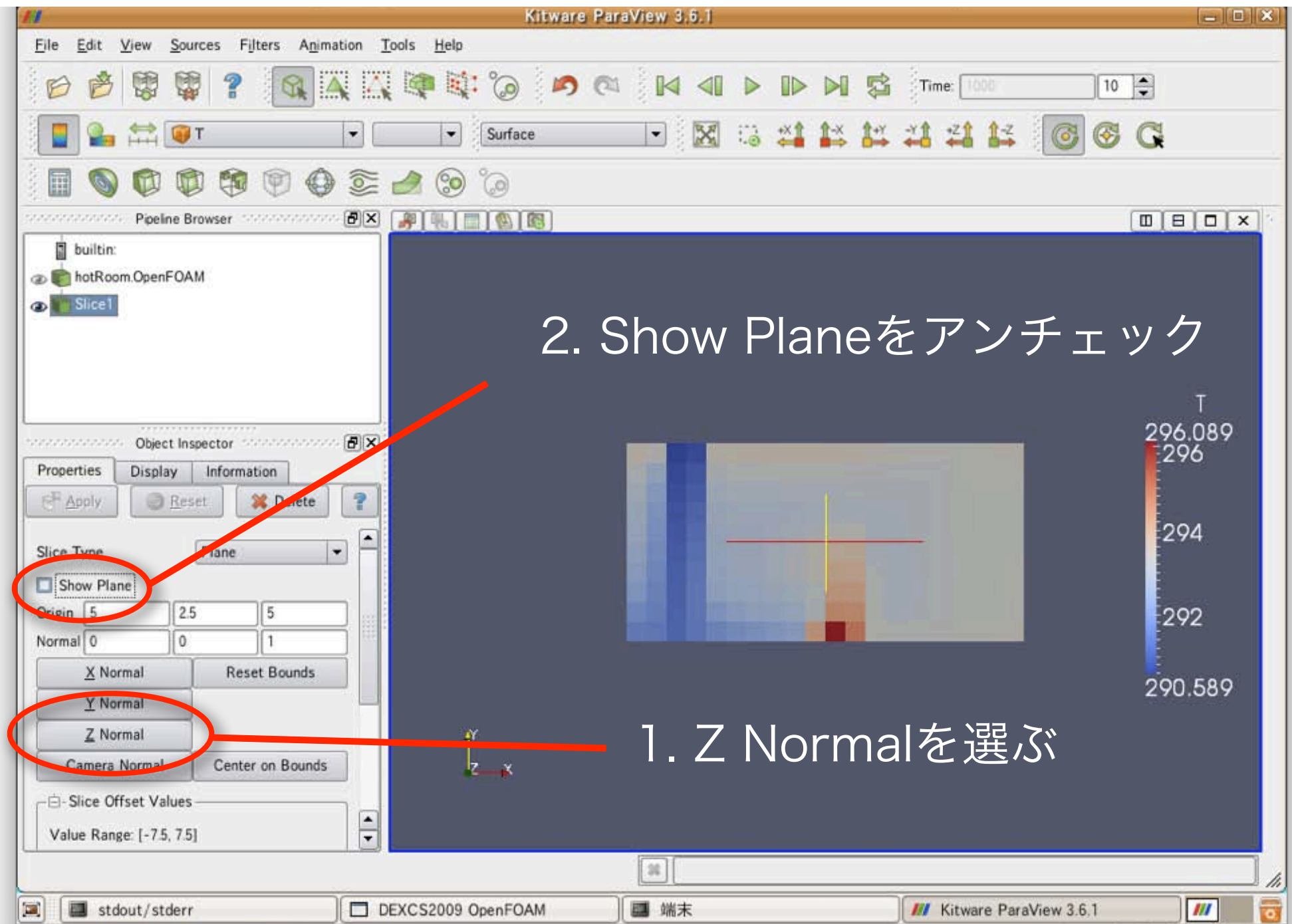
```
paraFoam & ↵
```

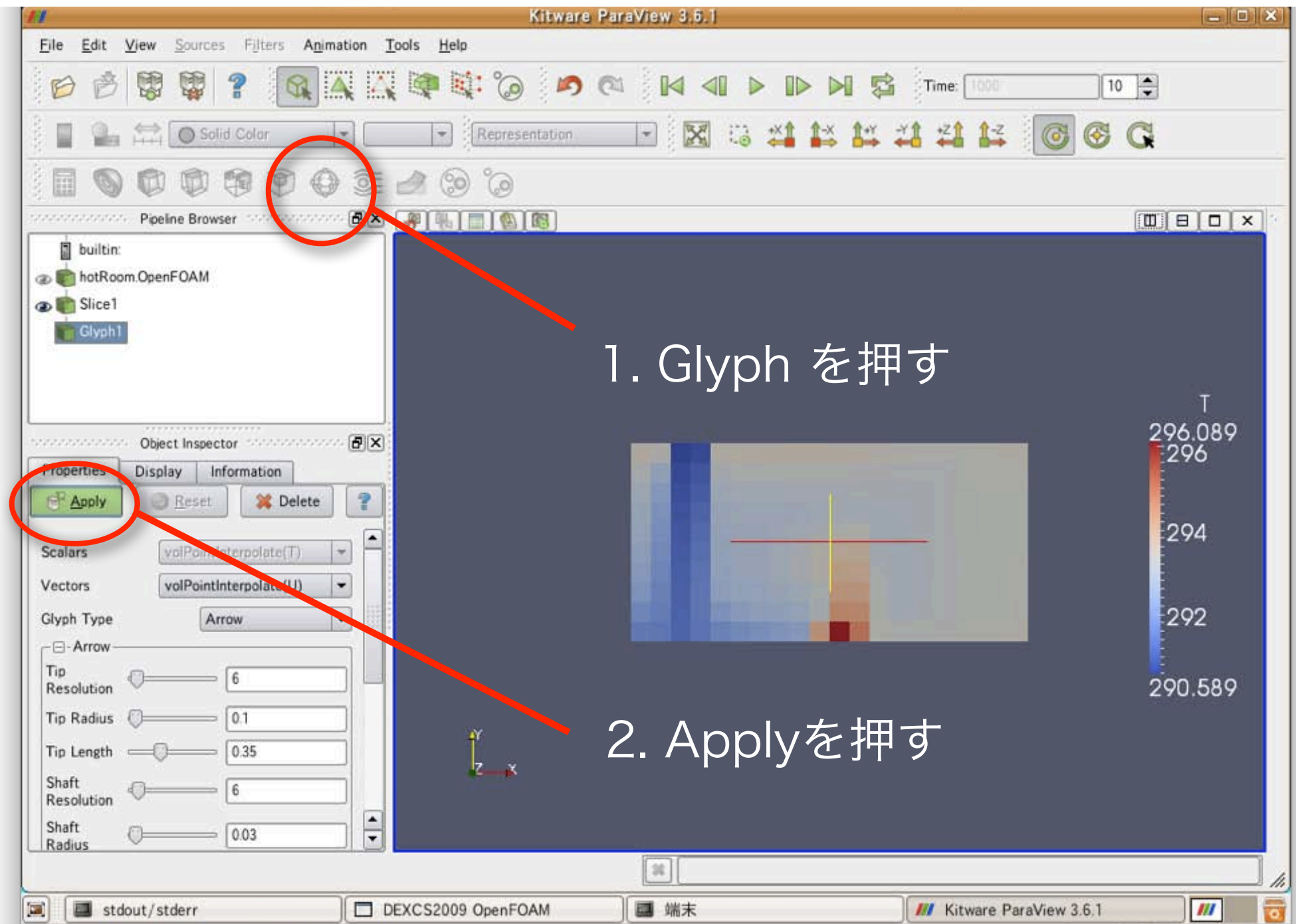


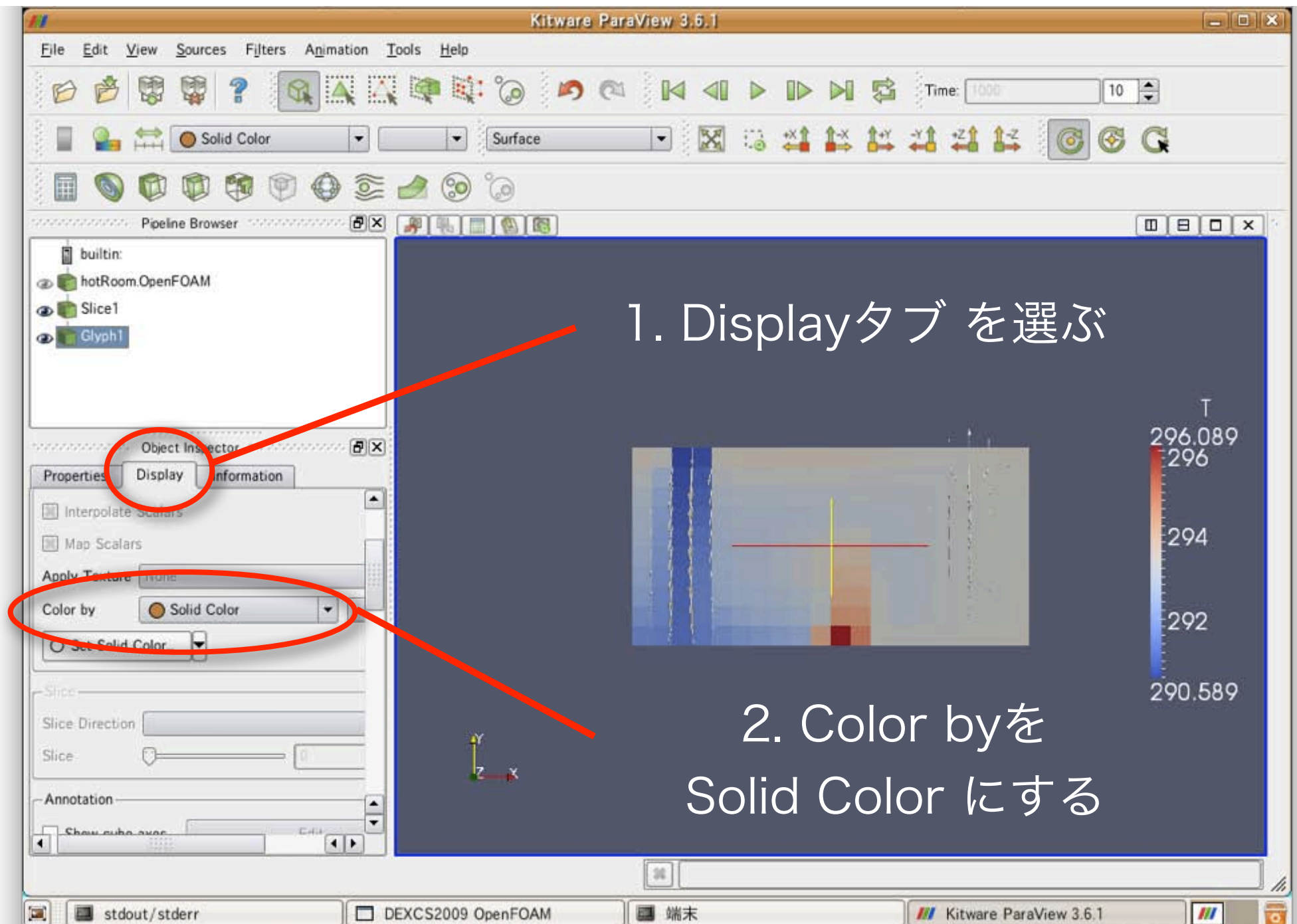


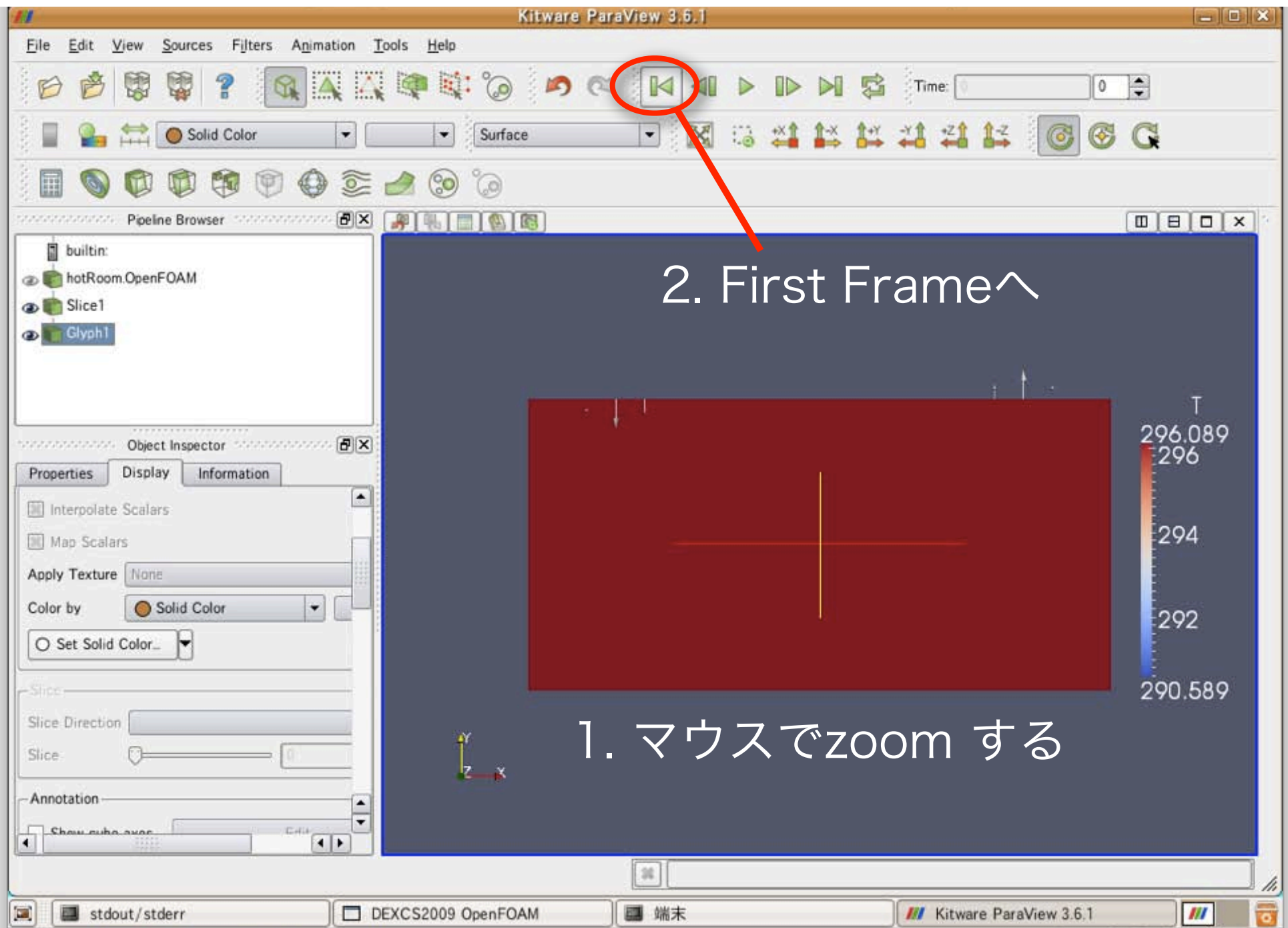


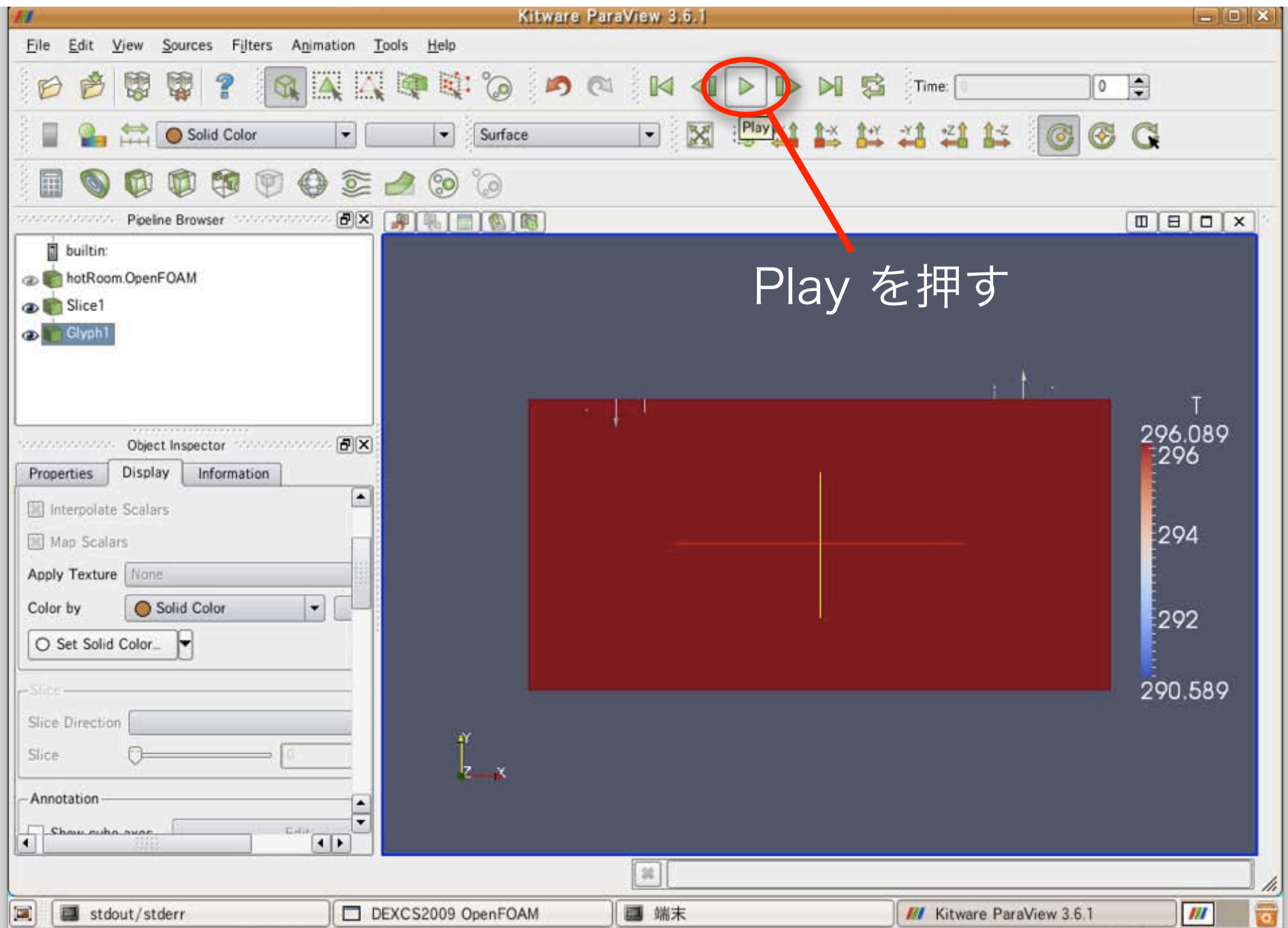


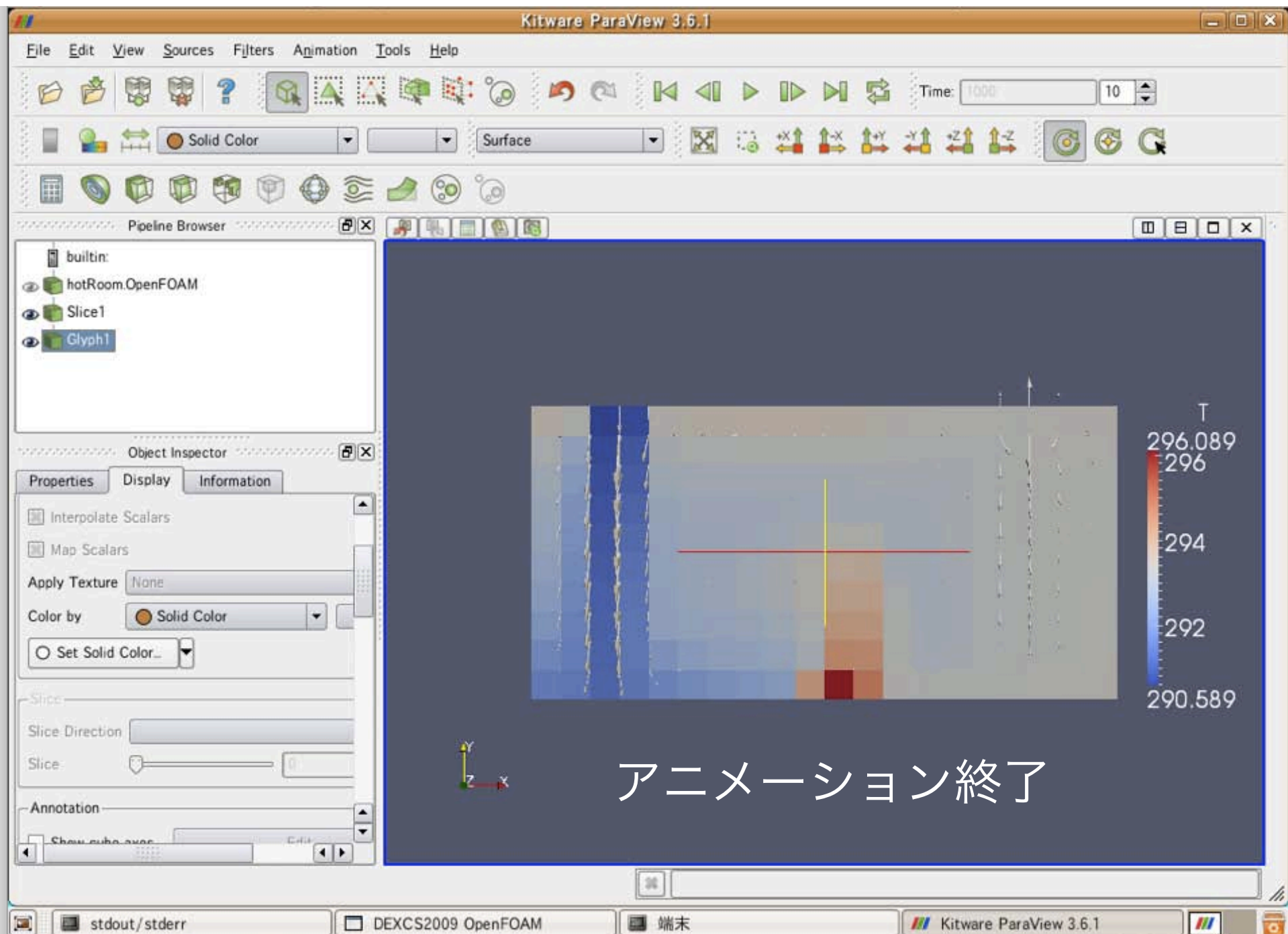












まとめ

1. 下準備
2. 発熱面に対応する新たなパッチ作成と初期値の設定
3. 吹出口・吸込口に対応する新たなパッチ作成と初期値の設定

今後の講習予定

1. snappyHexMeshによる格子生成
2. 並列計算
3. ソルバーのカスタマイズ