

OpenFOAM®非圧縮性流体解析演習シリーズ

第6回

blockMeshとsnappyHexMesh を用いた格子生成 (後編)

今野 雅 (オープンCAE学会、東京大学)

自己紹介

- 所属
 - 東京大学 大学院工学系研究科 建築学専攻
- 専門
 - 建築環境工学 (温熱・空気環境、特に数値予測)
- 所属学会
 - 日本建築学会
 - 空気調和・衛生工学会
 - 日本流体力学会
 - 日本風工学会
 - オープンCAE学会(副会長)



目次

1. 事前準備
2. snappyHexMeshの設定(後編)
3. snappyHexMeshの演習(後編)
4. 質疑



事前準備

端末の起動

The image shows a Linux desktop environment with a dark theme. The application menu is open, showing categories like 'アプリケーション', 'アクセサリ', 'インターネット', 'オフィス', 'グラフィックス', 'ゲーム', 'サウンドとビデオ', 'システムツール', 'プログラミング', and 'Ubuntuソフトウェアセンター'. The 'アプリケーション' category is circled in red, and the '端末(OF-1.7.x)' application is also circled in red. A callout box points to the terminal window with the text '2. 端末(OF-1.7.x)'. Another callout box points to the 'アプリケーション' category with the text '1. アプリケーション->DEXCS'. The terminal window is titled '端末' and shows the following text:

```
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
dexcs@dexcs-desktop:~$
```

A callout box below the terminal window contains the text: '端末が表われる (OpenFOAM-1.7.x用 環境設定済み)'. The OpenCAE logo is visible in the bottom right corner.

チュートリアルの初期化

```
run ↵
```

```
cd tutorials/ ↵
```

```
cd heatTransfer/ ↵
```

```
cd buoyantBoussinesqSimpleFoam/ ↵
```

```
cd iglooWithFridges/ ↵
```

```
foamCleanTutorials ↵
```

```
ls ↵
```

←ディレクトリ名・コマンド名はTabキーで補完できます

出力

```
0 Allrun constant system
```



OpenCAE

チュートリアルの実行

snappyHexMeshDictの修正

```
gedit system/snappyHexMeshDict &↵
```

```
debug 1; 途中経過の格子を出力するように変更
```

格子生成とsnappyHexMeshのログの確認

```
blockMesh↵ ベース格子生成
```

```
foamJob -s snappyHexMesh↵ ログを見ながら記録
```

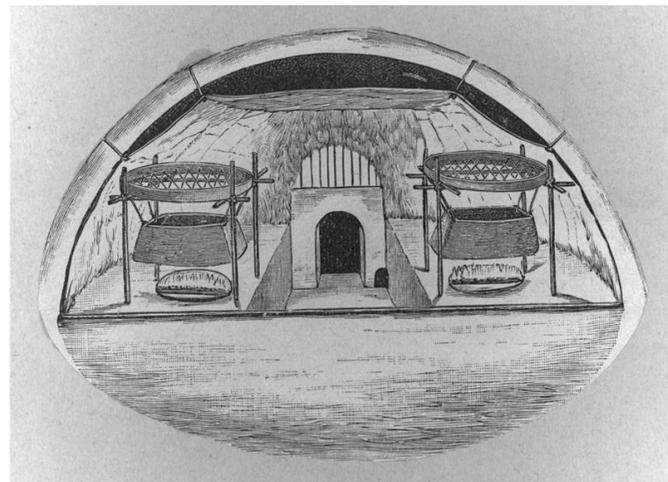


snappyHexMeshの設定 (後編)

iglooWithFridgesチュートリアル

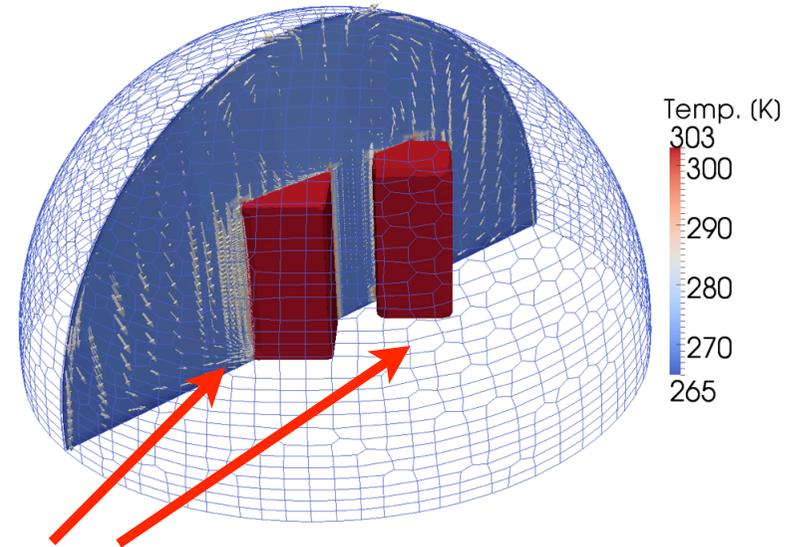


外観



内部

igloo(イグルー)とは：
カナダのエスキモー族イヌイット
が狩猟先で雪を用いて作る仮住居

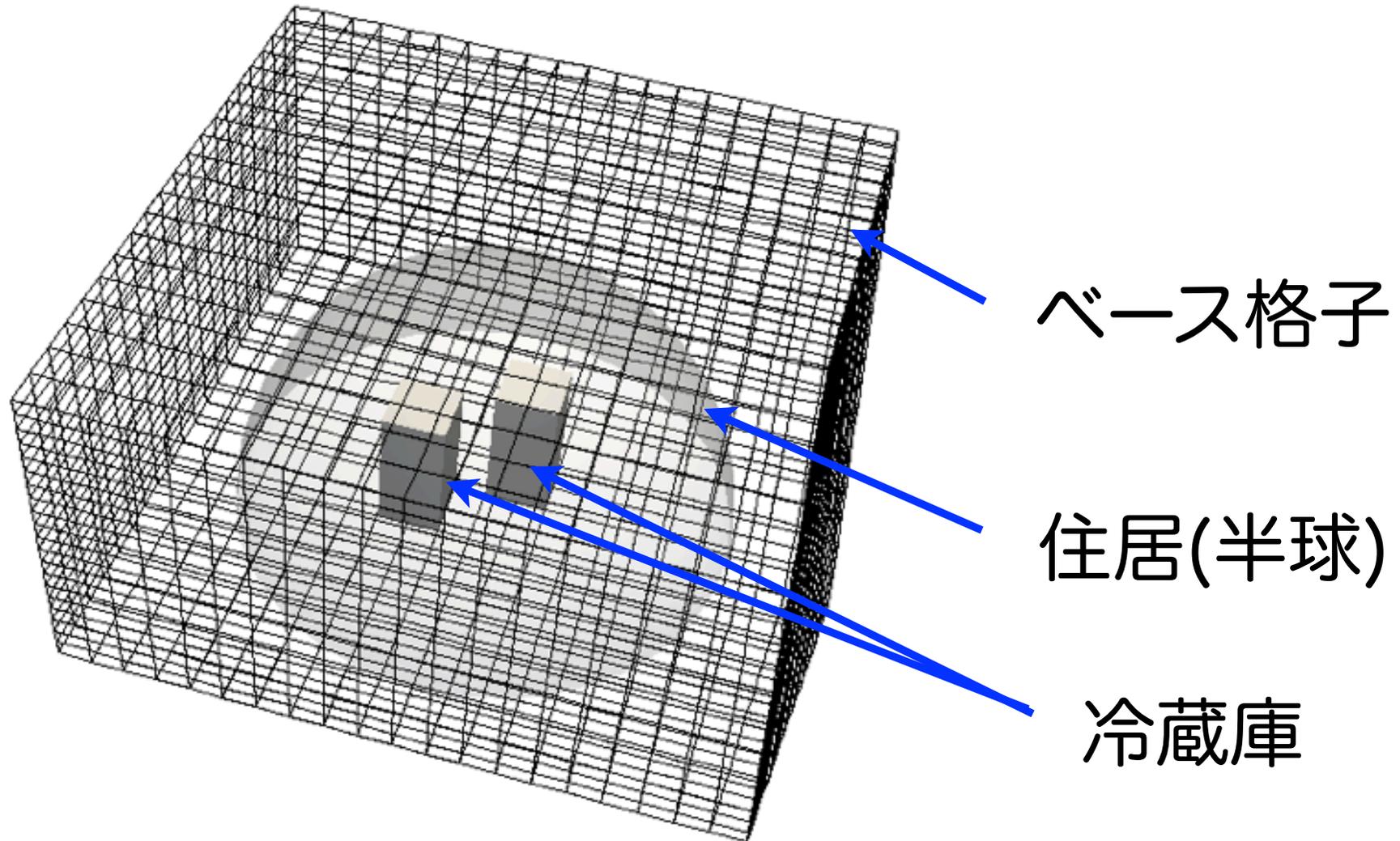


冷蔵庫?の発熱によるイグルー内
の熱対流を解く

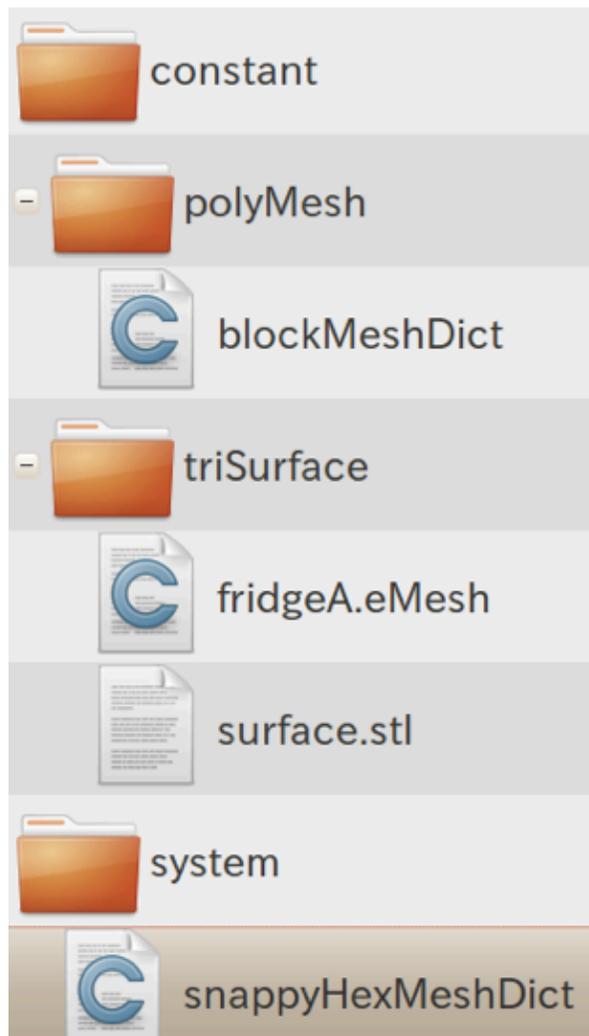
(図引用元: <http://ja.wikipedia.org/wiki/%E3%82%A4%E3%82%B0%E3%83%AB%E3%83%BC>)



ベース格子および幾何形状



格子生成の設定ファイル



…blockMeshの設定(ベース格子を定義)

…STLファイルや辺ファイルの置き場所

…特徴線の定義ファイル

…表面・領域ベース細分割用のSTL

…snappyHexMeshの設定

snappyHexMeshDict 1

階段状格子

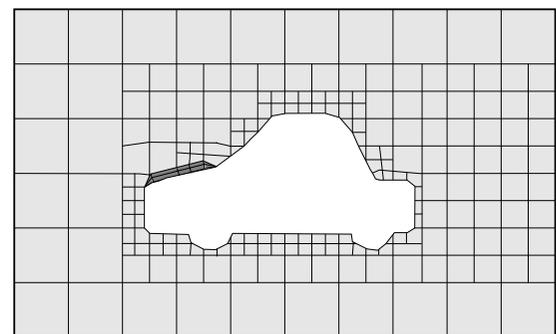
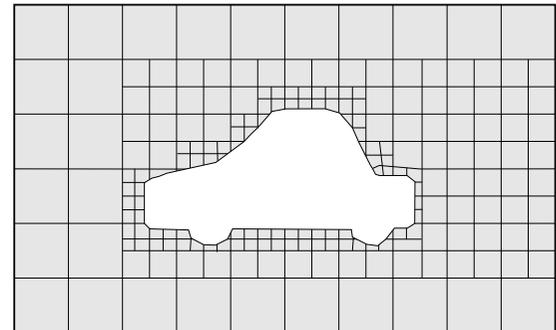
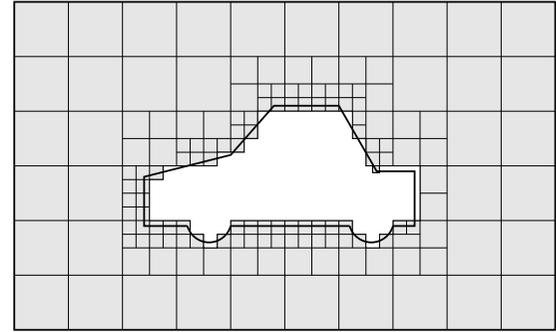
```
castellatedMesh true;
```

境界適合

```
snap true;  
しない場合は false;
```

レイヤ付加

```
addLayers true;  
しない場合は false;
```



snappyHexMeshDict 2

geometry 各種形状の定義 (前回説明)

castellatedMeshControls 階段状格子の制御設定 (前回説明)

snapControls 境界適合の制御設定

addLayersControls レイヤ付加の制御設定

meshQualityControls 格子品質の制御設定



meshQualityControls

生成格子品質の制御設定

```
maxNonOrtho 65;  
maxBoundarySkewness 20;  
maxInternalSkewness 4;  
maxConcave 80;  
minFlatness 0.5;  
minVol 1e-13;  
minArea -1;  
minTwist 0.05;  
minDeterminant 0.001;  
minFaceWeight 0.05;  
minVolRatio 0.01;  
minTriangleTwist -1;  
nSmoothScale 4;  
errorReduction 0.75;  
relaxed {maxNonOrtho 75;}
```

基本的には修正は不要

詳細snappyHexMeshDict
のコメントやOpenFOAM
ユーザガイドを参照



OpenCAE

snapControls

境界適合の制御設定

nSmoothPatch 3;

境界におけるパッチの平滑化(smoothing)の反復回数

tolerance 4.0;

表面形状の特徴的な点や辺に引き寄せられる頂点への最大距離の比

局所的な辺の最大長とこの比の積が実際の最大距離

nSolveIter 30;

格子の移動時の緩和計算の回数

nRelaxIter 5;

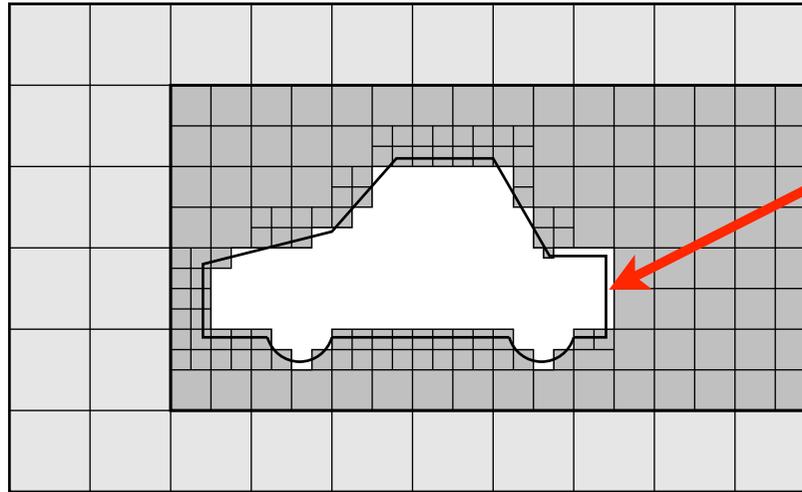
格子の表面への適合時の緩和計算の最大回数

基本的には修正は不要



境界適合プロセス 1

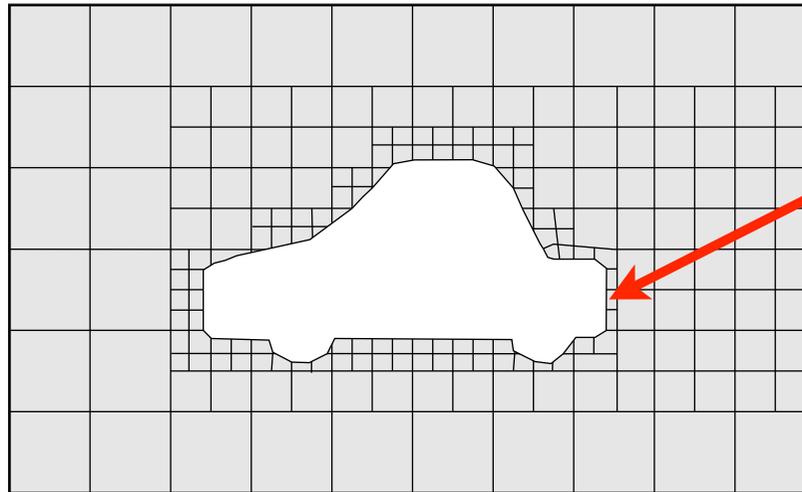
階段状
格子



STL形状
または
基礎形状

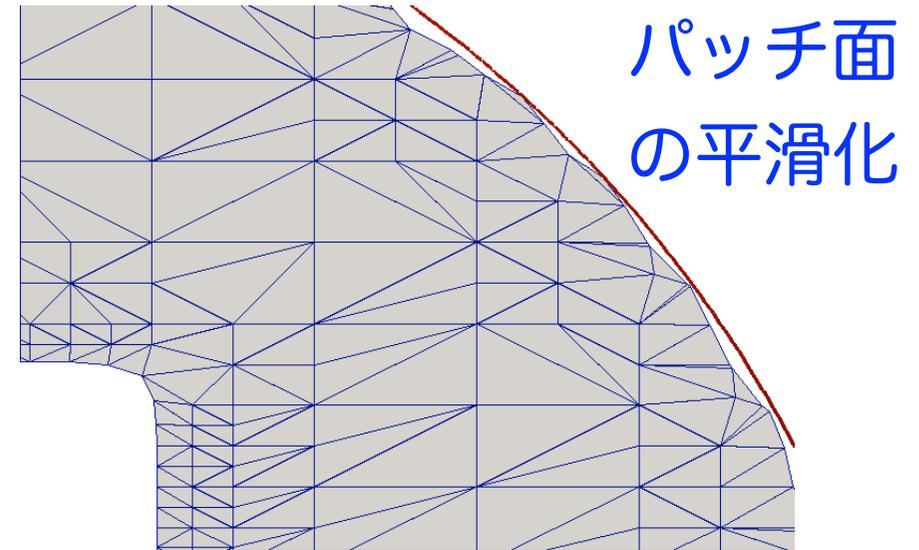
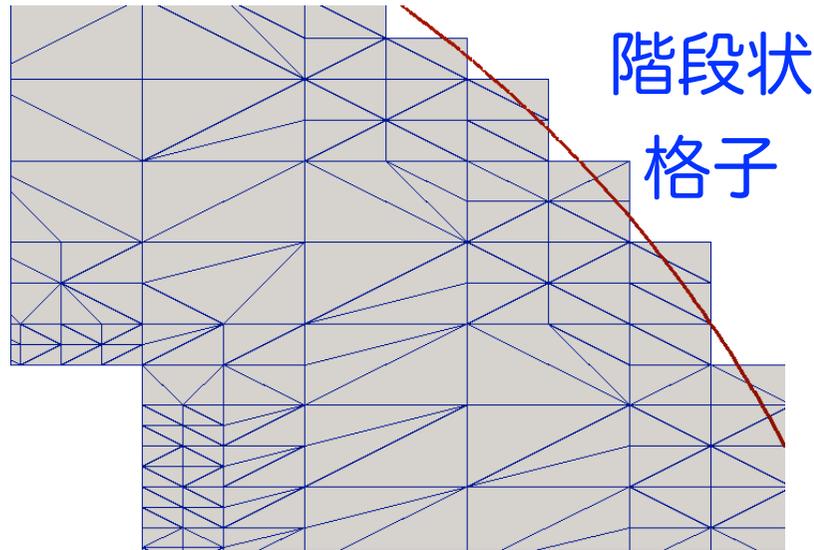


境界適合
格子

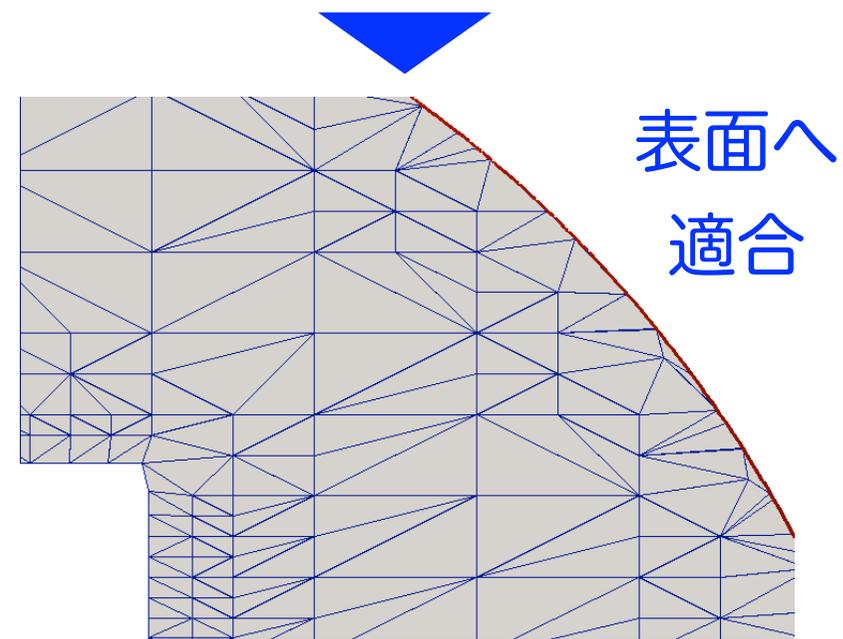


形状周辺の
格子を滑ら
かに変形さ
せて適合

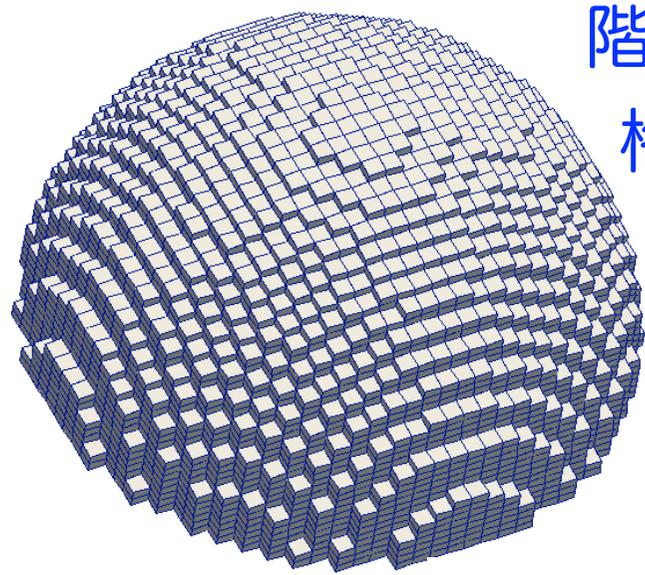
境界適合プロセス 2



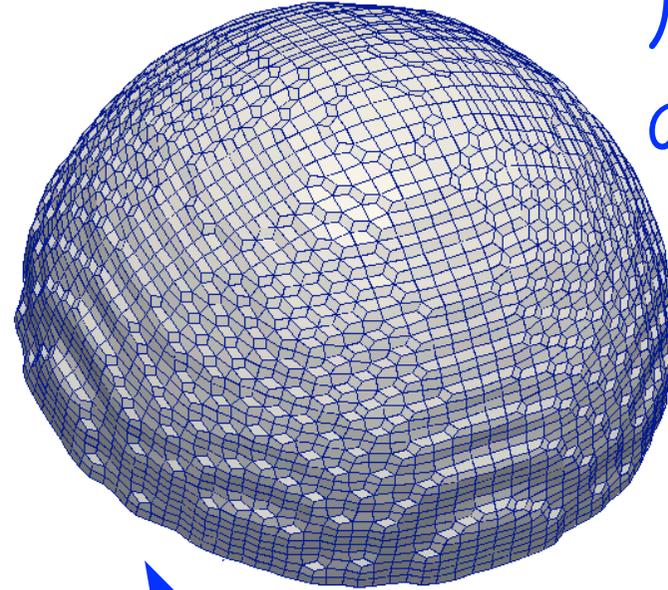
1. 階段状格子のパッチ面を平滑にする(smoothing)。
2. 境界上の頂点を表面形状に移動させ、格子品質を検査
3. 品質が制限値を満すまで移動を緩和して反復する。



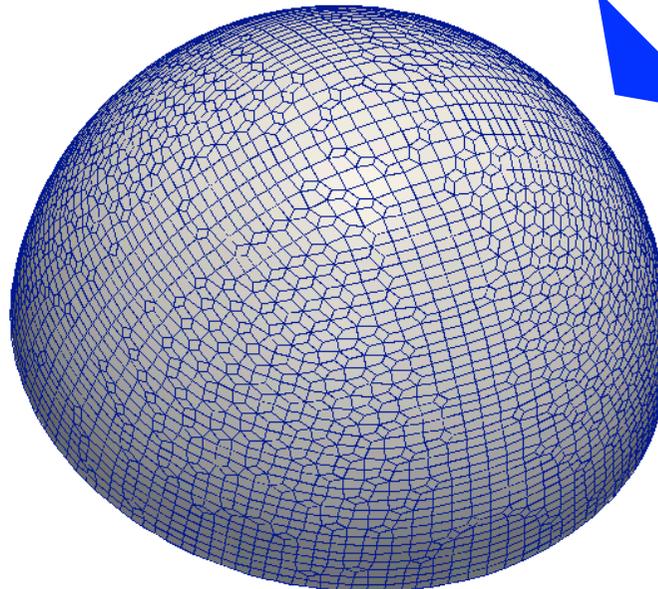
境界適合プロセス3



階段状
格子

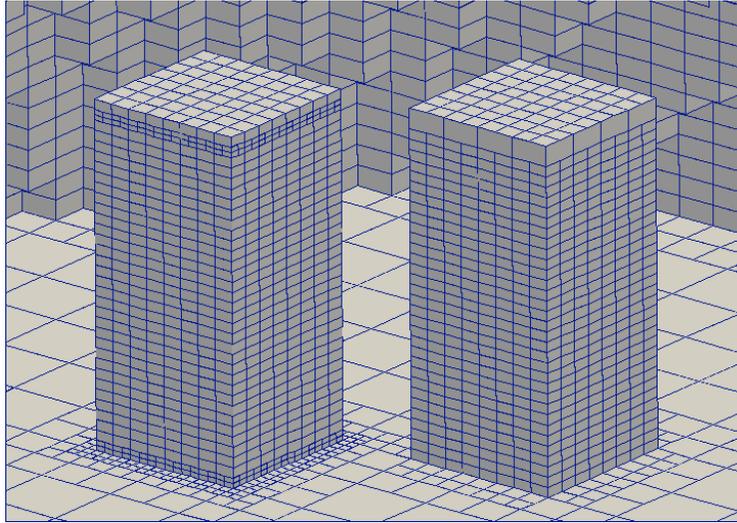


パッチ面
の平滑化

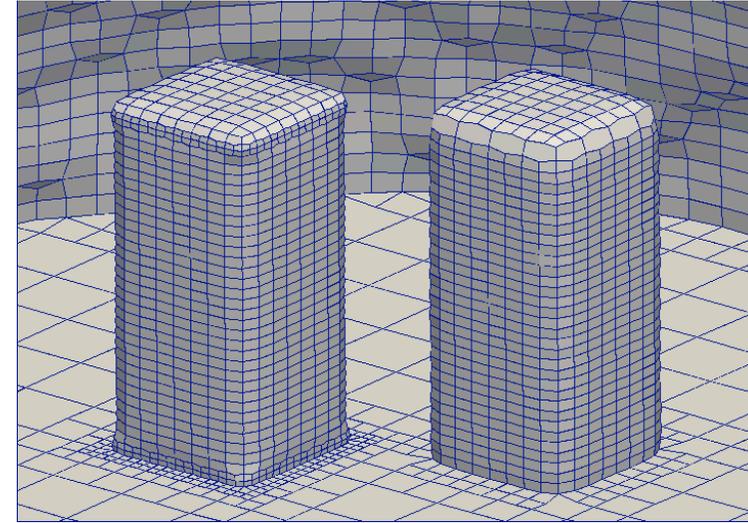


表面へ
適合

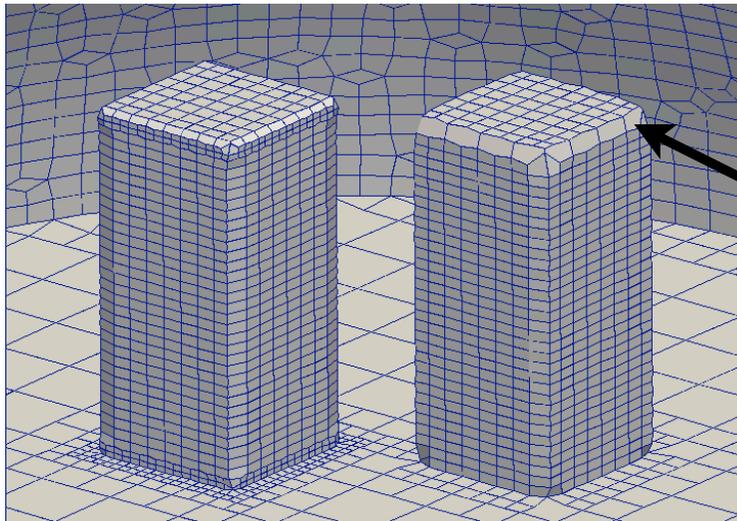
境界適合プロセス4



階段状格子



パッチ面の平滑化



表面へ適合

特徴線の再現機能は無いので、特徴線は丸くなる

addLayersControls 1

relativeSizes true; true/false

レイヤ厚さをレイヤ外部の変形していない格子の大きさに対する相対値とするか(true)、または絶対値とするか(false)

expansionRatio 1.0;

レイヤメッシュの拡大比(r)

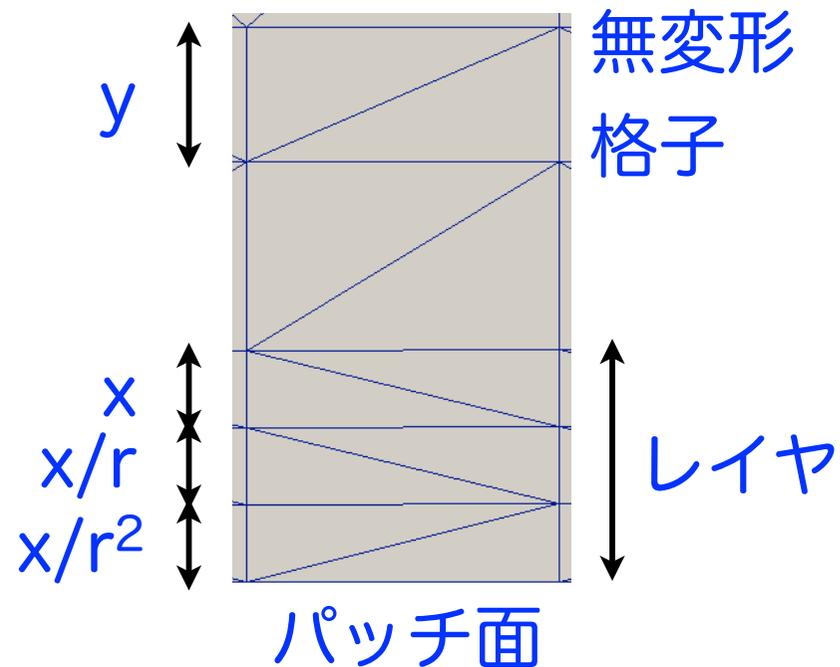
finalLayerThickness 0.5;

壁から最も遠い層の厚さ(x または

relativeSizes true $\rightarrow x/y$)

minThickness 0.25;

格子のレイヤの最小の厚さ

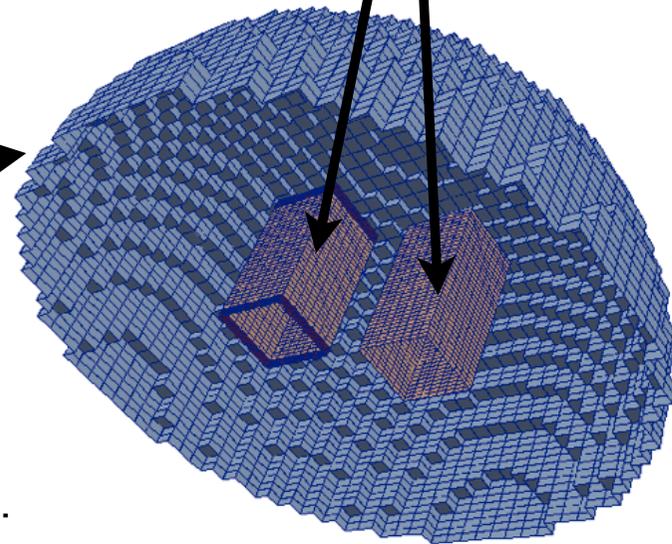


これ以外は基本的には修正不要

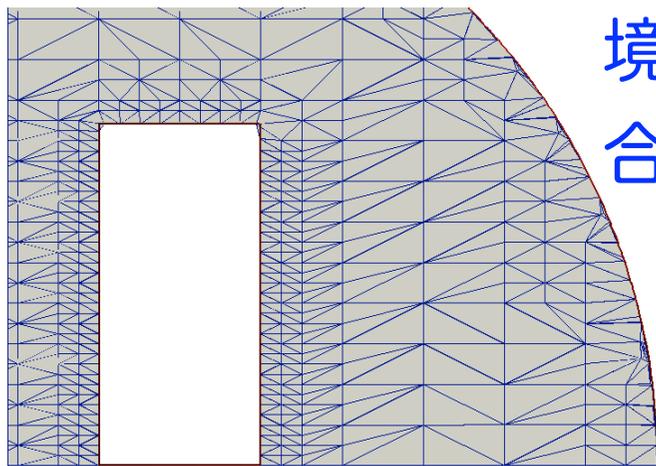
addLayersControls 2

layers 付加するレイヤの定義

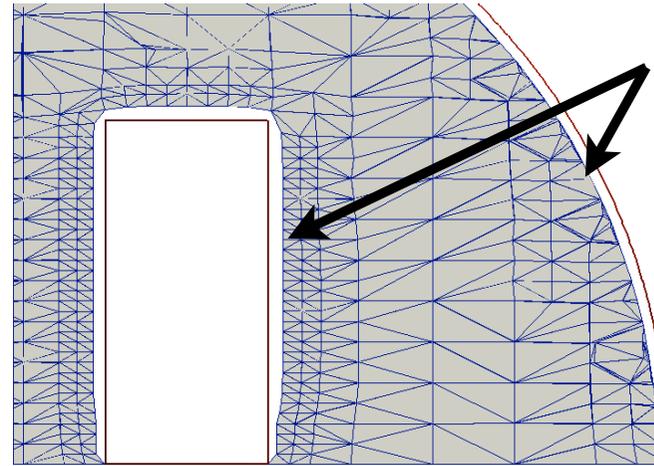
```
{  
  "two.*" パッチ名 (“正規表現も可”)  
  → twoFridgeFreezers_seal_0, twoFridgeFreezers_herring_1  
  {  
    nSurfaceLayers 3; 付加するレイヤ数  
  }  
  
  "igloo_.*" パッチ名  
  → igloo_region0  
  {  
    nSurfaceLayers 1;  
  }  
}
```



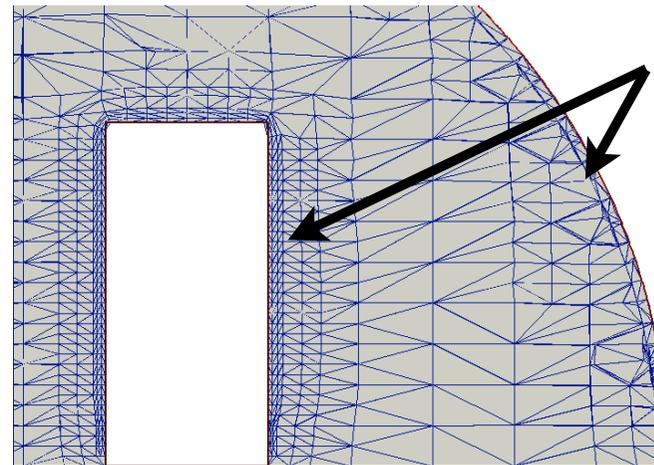
レイヤ付加プロセス 1



境界適
合格子



パッチ面
の後退
(shrink)



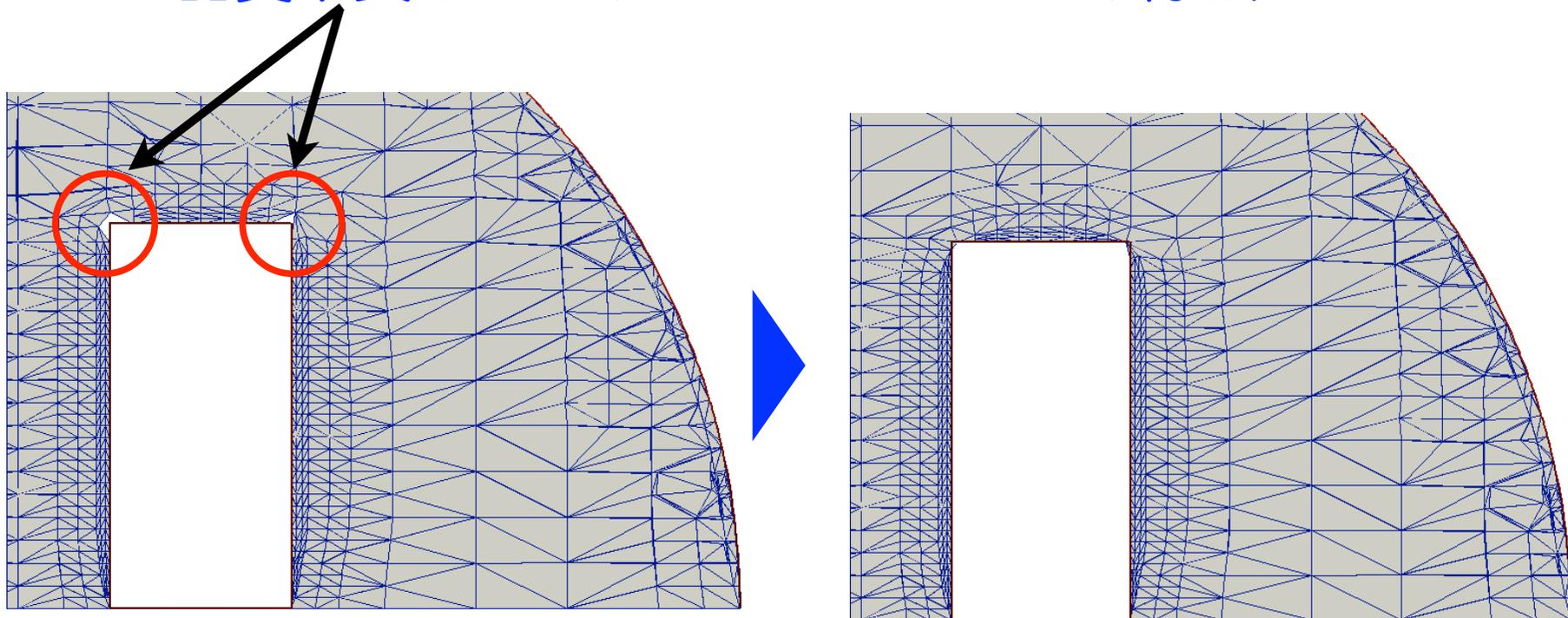
3層と
1層のレ
イヤ挿入

格子品質を検査し、
不良なレイヤは除去
して、shrinkに戻り
反復する

レイヤ付加プロセス 2

品質不良なレイヤ

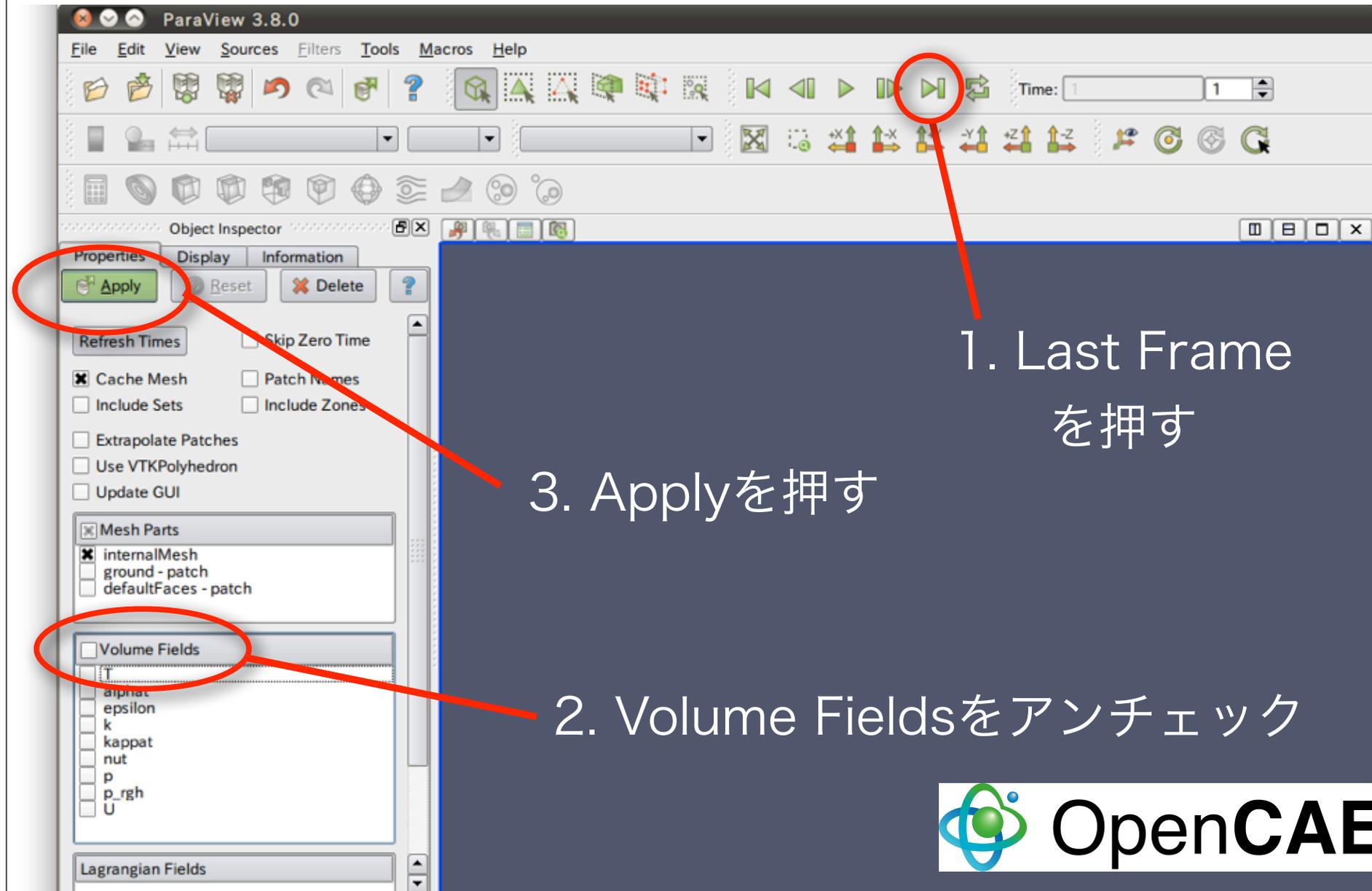
レイヤ除去





snappyHexMeshの演習 (後編)

paraFoamによる格子可視化 1



The screenshot shows the ParaView 3.8.0 interface. The 'Object Inspector' panel on the left has three red circles highlighting the 'Apply' button, the 'Volume Fields' section, and the 'internalMesh' checkbox. The 'Volume Fields' section is currently unchecked. The 'Apply' button is also highlighted with a red circle. A red arrow points from the 'Apply' button to the 'Last Frame' button in the top toolbar. A blue box contains the following text:

1. Last Frame を押す
2. Volume Fieldsをアンチェック
3. Applyを押す

The OpenCAE logo is visible in the bottom right corner of the blue box.

paraFoamによる格子可視化 2

The image shows a screenshot of the ParaView 3.8.0 software interface. The main window displays a 3D model of a sphere. The interface includes a menu bar (File, Edit, View, Sources, Filters, Tools, Macros, Help), a toolbar with various icons, and a Pipeline Browser on the left. The Object Inspector on the right shows the 'Slice1' filter selected. The 'Properties' panel for 'Slice1' is visible, showing the 'Slice Type' set to 'Plane', the 'Show Plane' checkbox unchecked, and the 'Origin' set to 3.5. The 'Normal' vector is set to (0, 1, 0), and the 'Y Normal' button is highlighted. The 'Apply' button is also highlighted. Red circles and arrows point to these elements, corresponding to the numbered instructions.

1. Clipを押す

2. Y Normalを押す

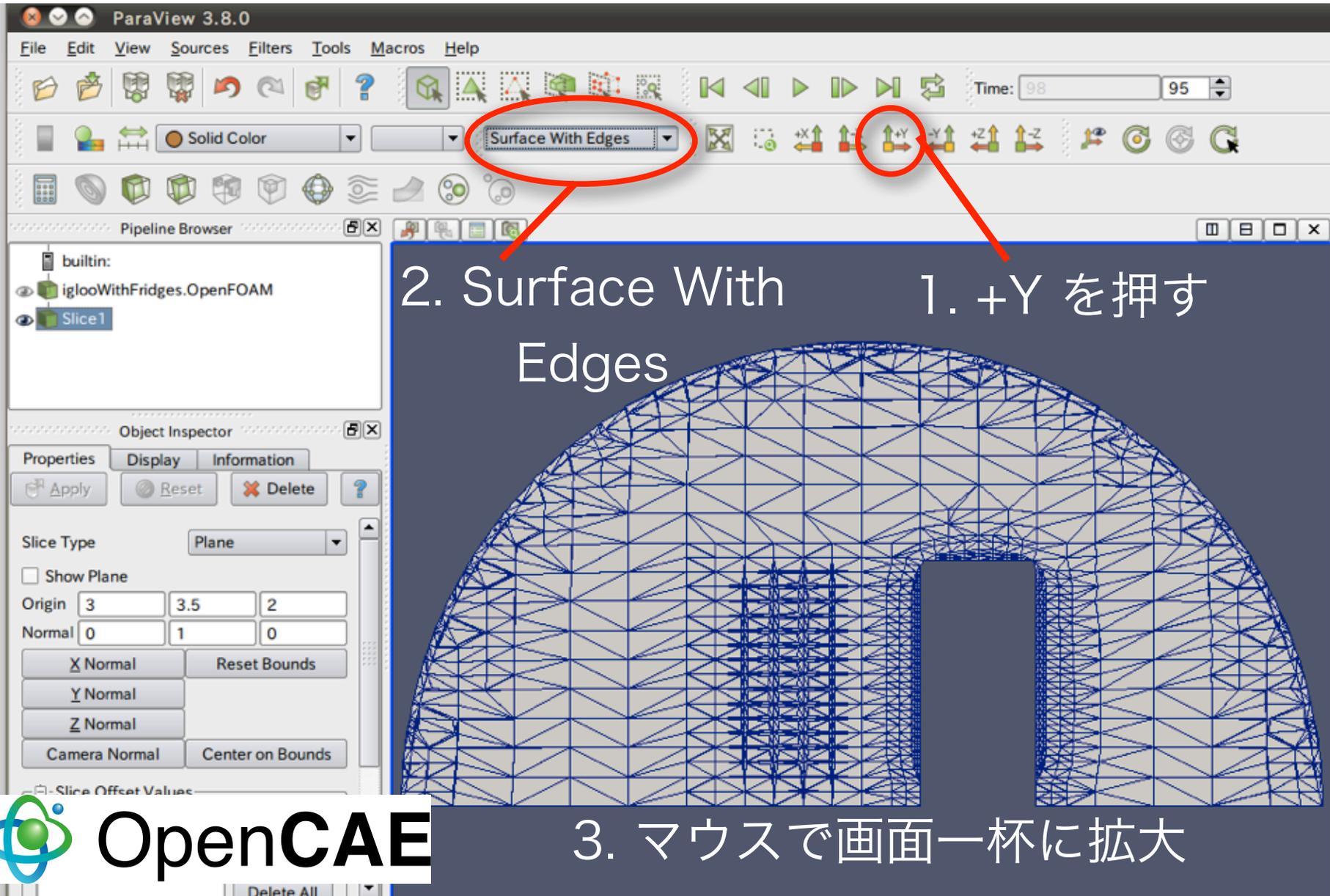
3. y座標に3.5を代入

4. Show Planeを外す

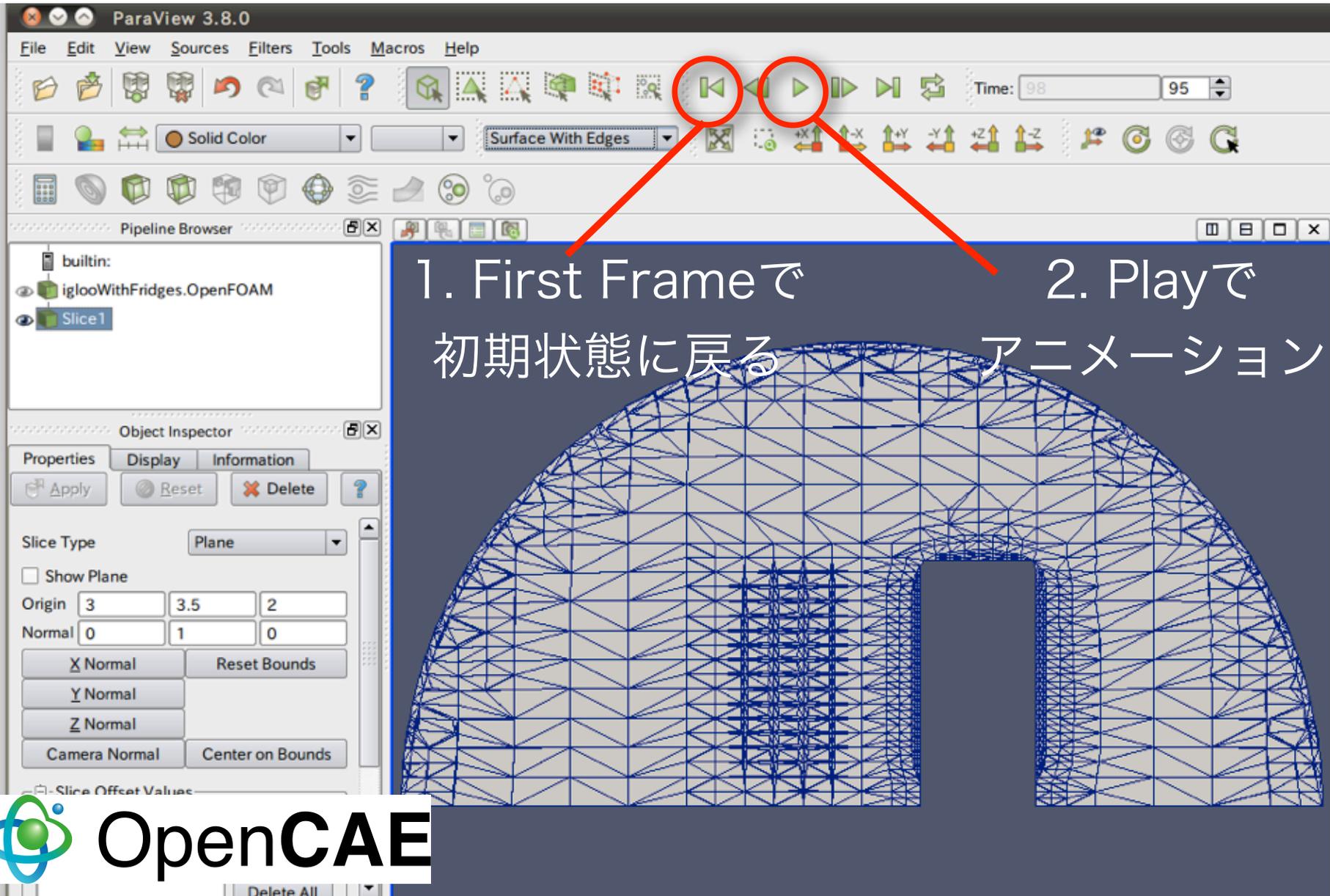
5. Apply

 **OpenCAE**

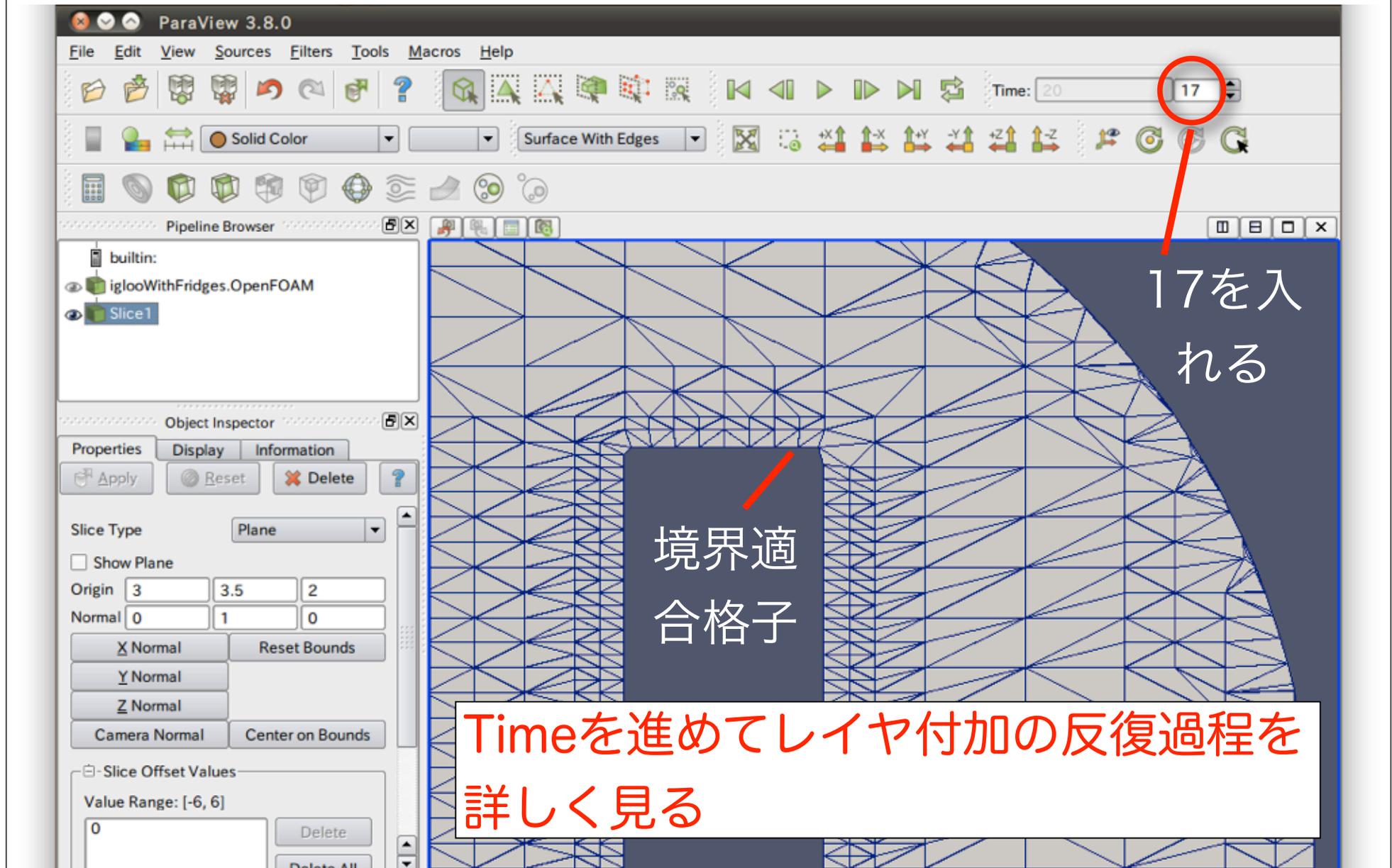
paraFoamによる格子可視化 3



paraFoamによる格子可視化 4



paraFoamによる格子可視化 5



演習1 (レイヤ数変更) 1

snappyHexMeshDictの修正 (geditなどで)

```
layers
{
  "two.*"
  {
    nSurfaceLayers 2; レイヤ数を3から2に変更
  }
}
(中略)
debug 0; 途中経過の格子を出力しないように戻す
```

snappyHexMeshの再実行 (端末で)

pyFoamClearCase.py . 格子を消す(最後にドット .が必要)

↑ 下の行が出るまで押す

foamJob -s snappyHexMesh ↵



OpenCAE

演習1 (レイヤ数変更) 2

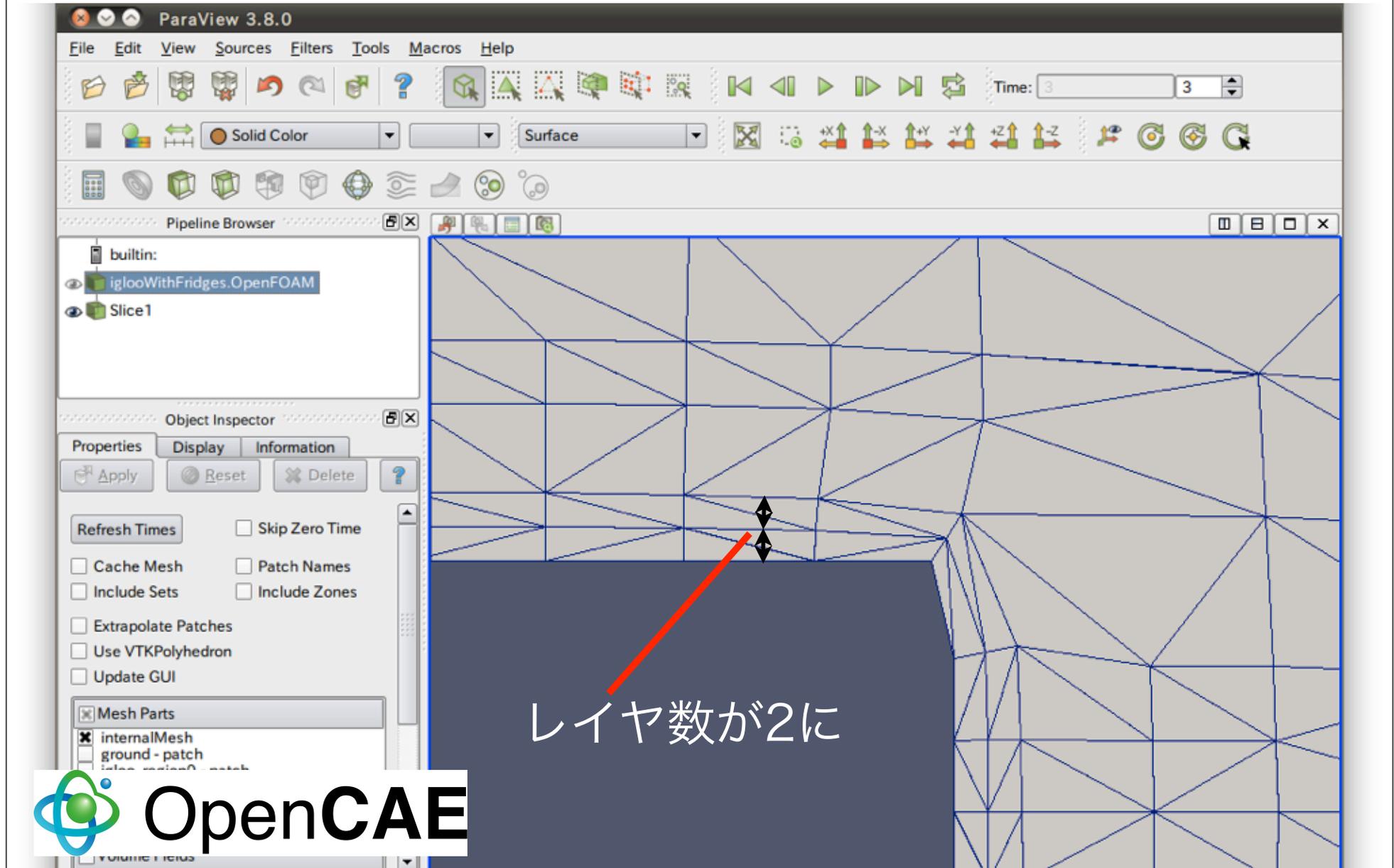
1. First Frameで初期状態に戻る

2. 格子が変更されているのでCache Meshをアンチェック

3. Last Frameで最終状態にする

OpenCAE

演習1 (レイヤ数変更) 3



演習2(メッシュ拡大比変更) 1

snappyHexMeshDictの修正 (geditなどで)

```
expansionRatio 1.5; レイヤメッシュの拡大比を1から1.5に
```

snappyHexMeshの再実行 (端末で)

↑ 下の行が出るまで押す

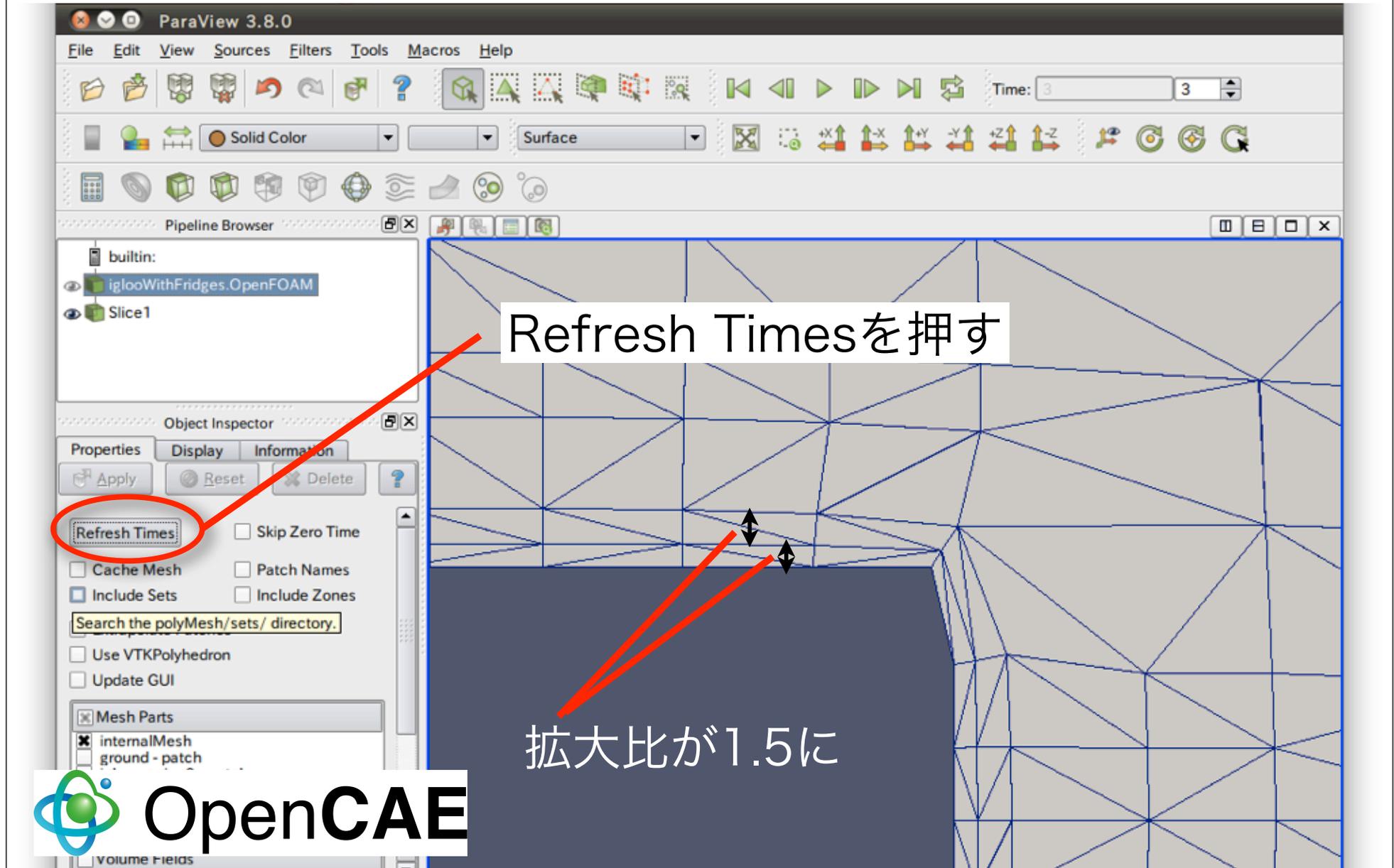
pyFoamClearCase.py . 格子を消す(最後にドット .が必要)

↑ 下の行が出るまで押す

```
foamJob -s snappyHexMesh ↵
```



演習2(メッシュ拡大比変更) 2



演習2(最終層厚み変更) 1

snappyHexMeshDictの修正 (geditなどで)

```
finalLayerThickness 1; 最終層厚みを外側の格子厚みと同じに
```

snappyHexMeshの再実行 (端末で)

↑ 下の行が出るまで押す

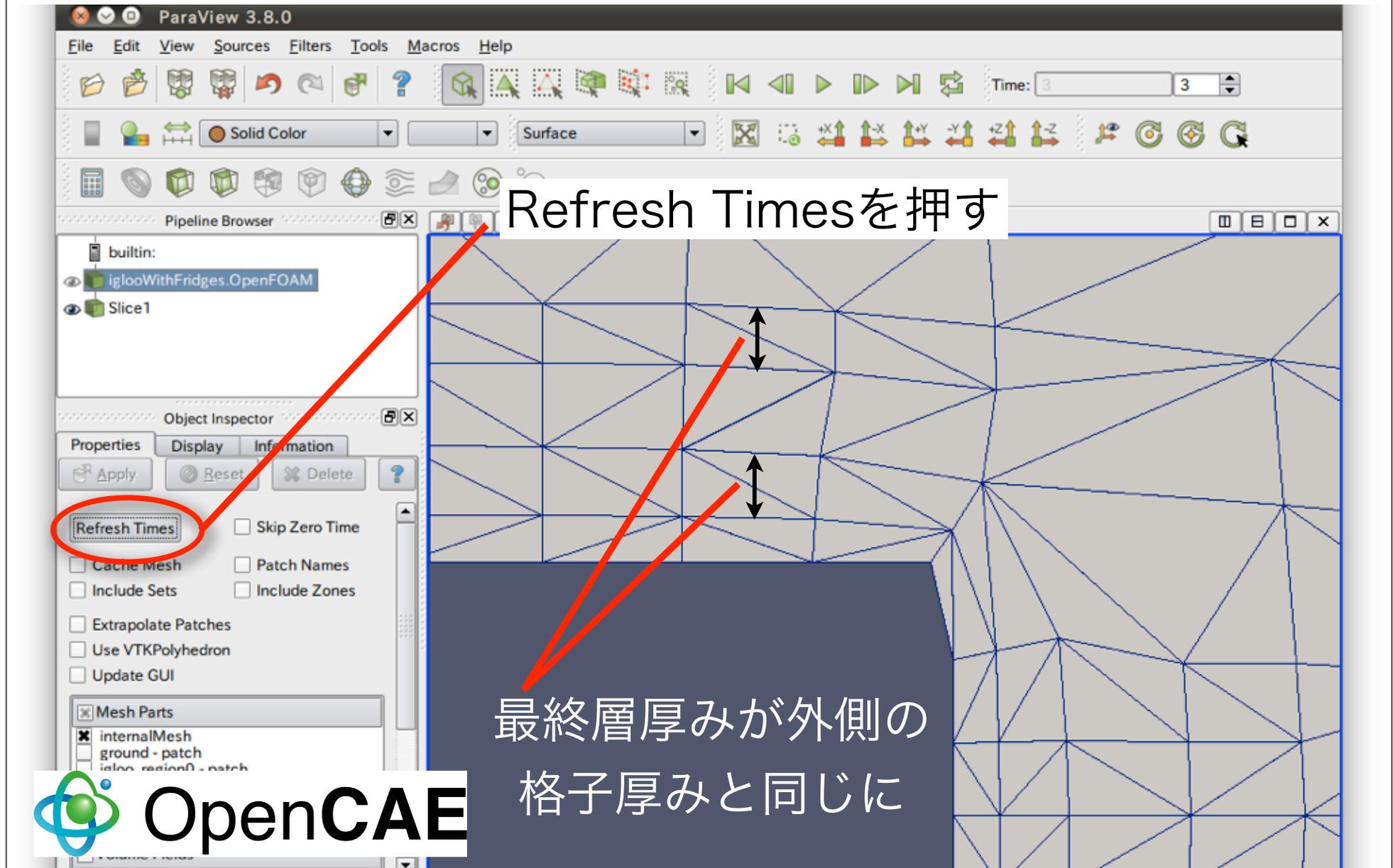
pyFoamClearCase.py . 格子を消す(最後にドット .が必要)

↑ 下の行が出るまで押す

```
foamJob -s snappyHexMesh ↵
```



演習2(最終層厚み変更) 2



目次

1. 事前準備
2. snappyHexMeshの設定(後編)
3. snappyHexMeshの演習(後編)
4. 質疑