平成22年度OpenFOAM®非圧縮性流体解析演習シリーズ

第5回 blockMeshとsnappyHexMeshを用いた格子生成 (前編)

今野 雅　（オープンCAE学会、東京大学）

OpenCAE

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)

OpenCAE

目次

1. snappyHexMeshの特徴
2. snappyHexMeshの設定(前編)
3. snappyHexMeshの演習(前編)
4. 質疑

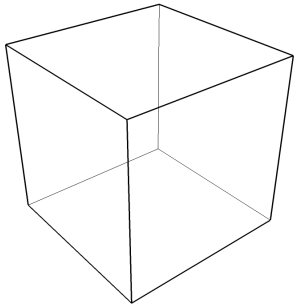
OpenCAE

snappyHexMeshの特徴

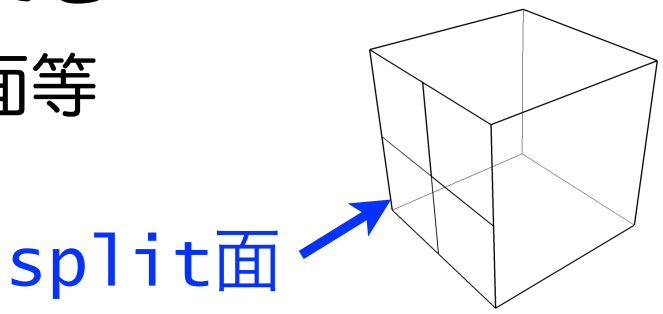
OpenCAE

snappyHexMeshの特徴 1

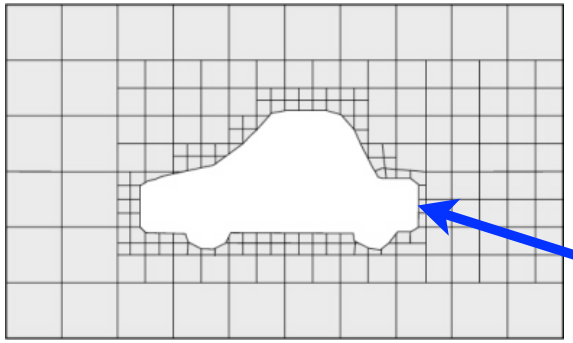
- 六面体格子(hex)または”面が分割された六面体的状格子”(split hex)からなる格子を自動生成する。
- STL表面に適合した格子が生成できる(辞書で定義される直方体や球面等の基礎形状も利用可)



hex



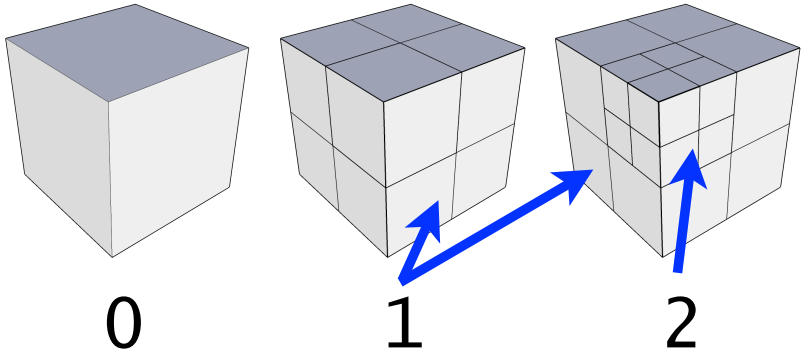
split hex

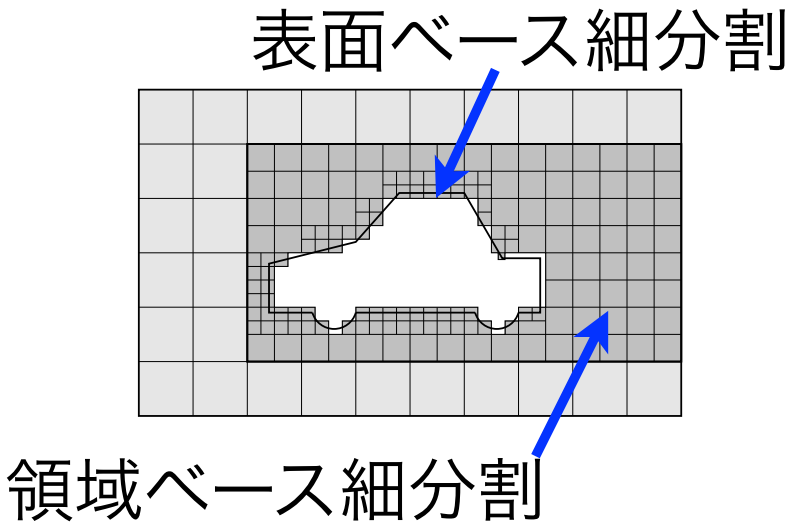


OpenCAE

snappyHexMeshの特徴 2

- 格子の細分割は8分木(2x2x2の分割を再帰的に行う)
- STL表面や基礎形状の表面の細分割レベルを指定できる(表面ベースの細分割)
- 細分割領域をSTL表面や基礎形状を用いて別途定義できる(領域ベースの再分割)

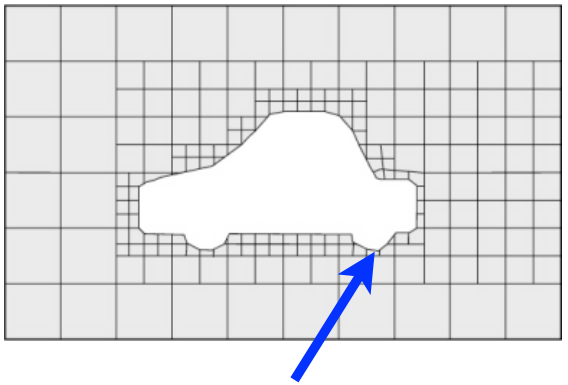




OpenCAE

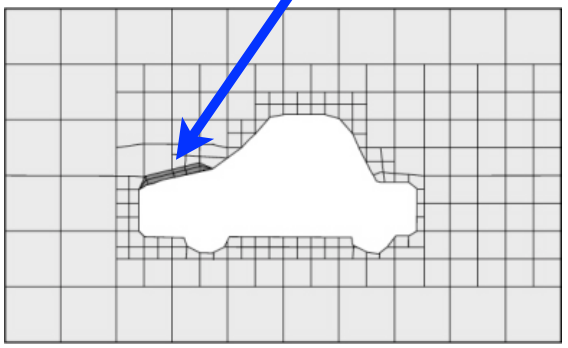
snappyHexMeshの特徴 3

- STL表面や基礎形状の表面に境界適合するように格子を滑らかに移動させることができる
- STL表面や基礎形状の表面にレイヤを滑らかに挿入できる



境界適合(snapping)

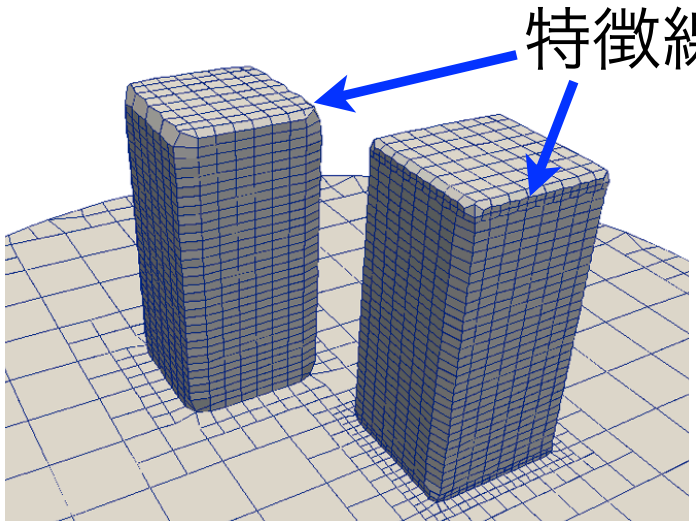
レイヤ挿入

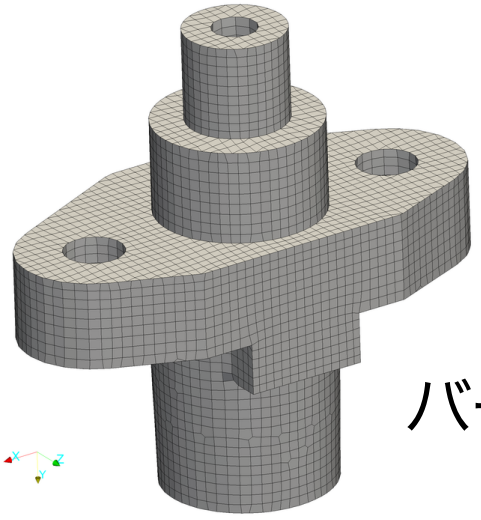


OpenCAE

snappyHexMeshの特徴 4

- 分割領域毎の格子が揃うように(ロードバランシング)並列に格子生成ができる
- 境界適合における特徴線の再現機能はver-1.7では実装されておらず、特徴線は丸くなる
- ただし、次期バージョンでは特徴線再現機能が実装されると4/11にOpenCFD社が告知

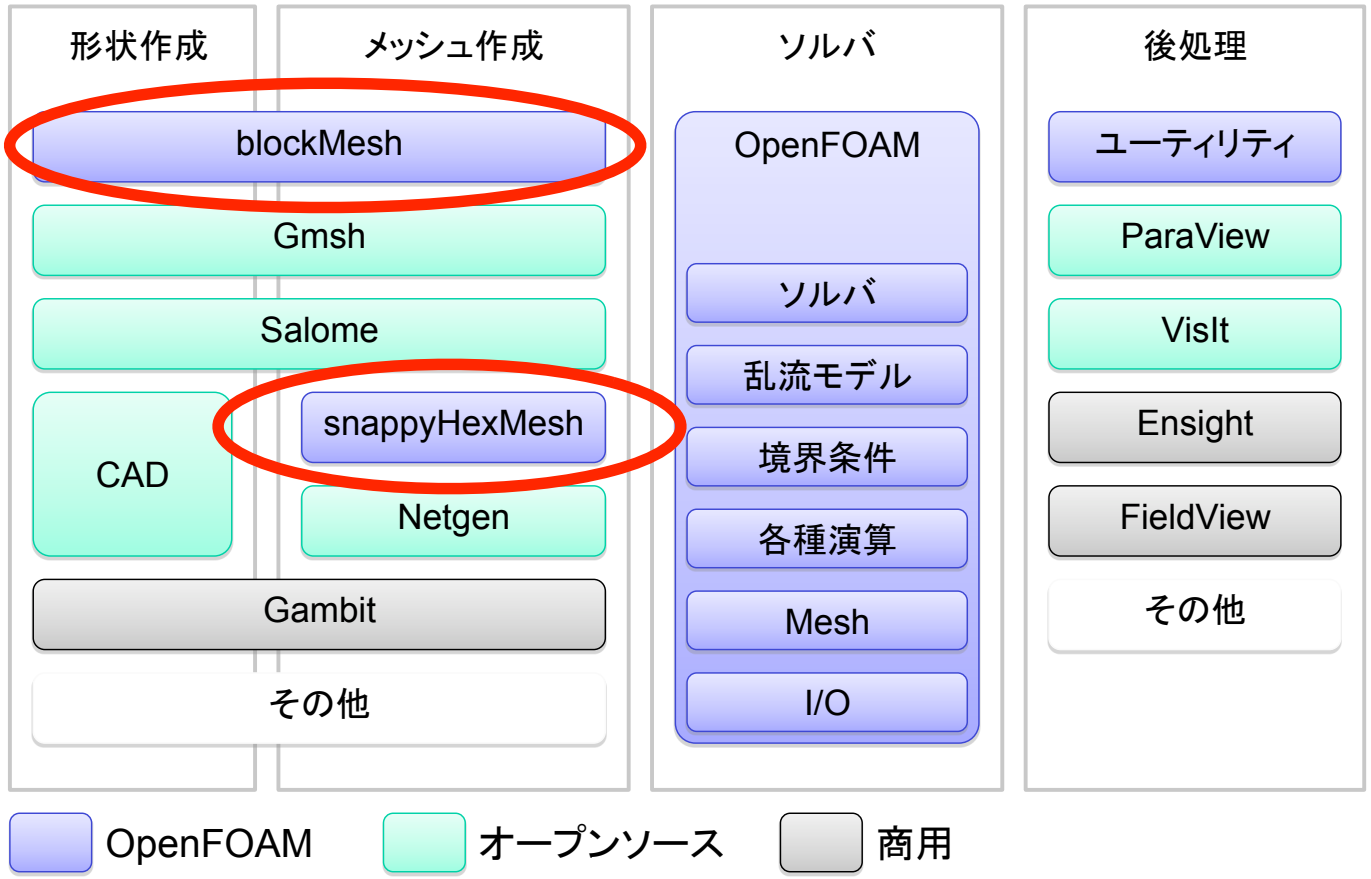




次期バージョン

OpenCAE

OFの格子生成のユーティリティ



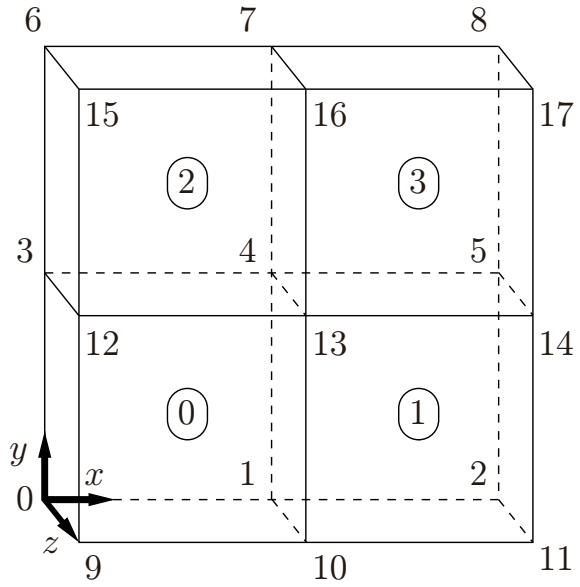
(図引用元: 大嶋 拓也 (新潟大学)「イントロダクション: OpenFOAM概要」
第一回OpenFOAM講習会)

OpenCAE

blockMesh

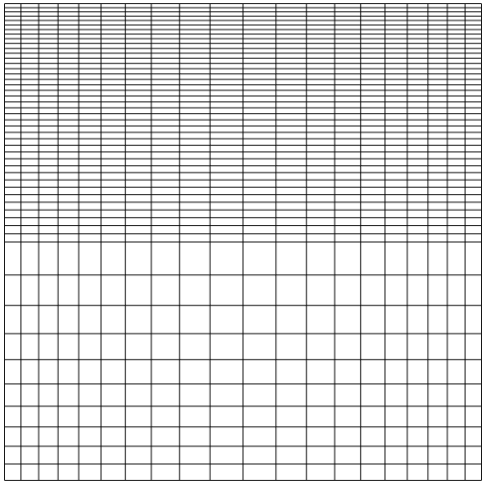
節点座標や分割方法を

blockMeshDictで定義



blockMesh

生成格子

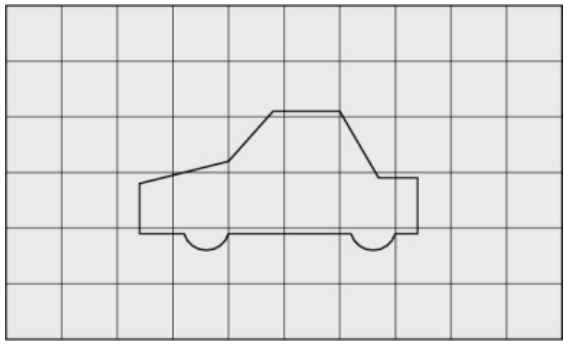


基礎的な六面体構造格子を生成する

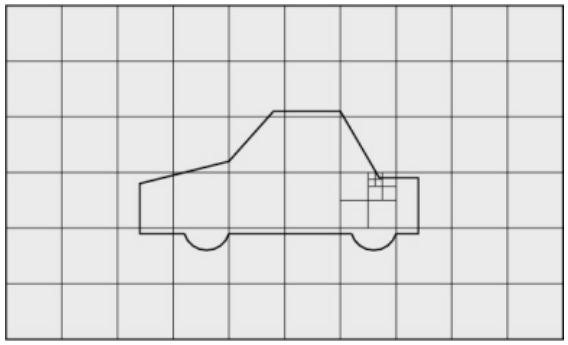
(図引用元: OpenFOAM User Guide Version 1.5)

OpenCAE

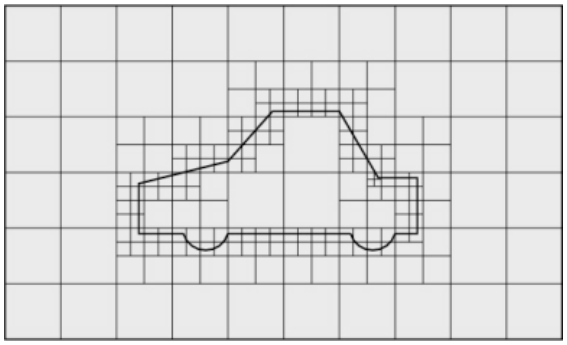
snappyHexMesh



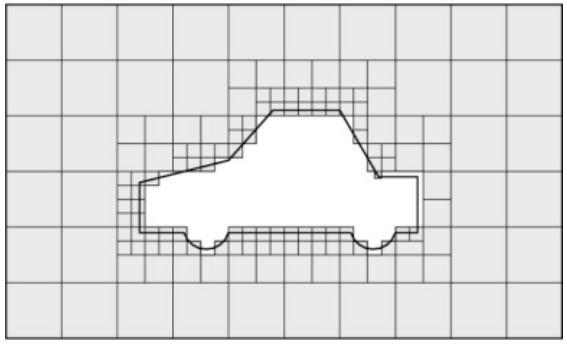
ベース六面体格子



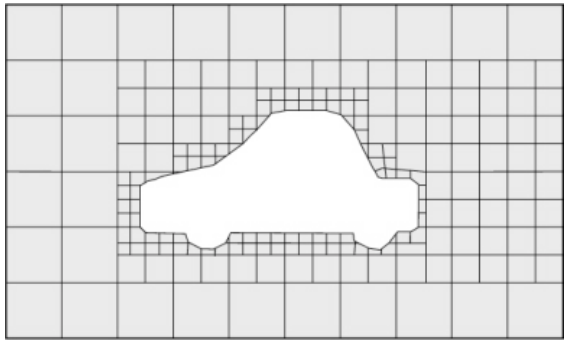
特徴線周辺分割



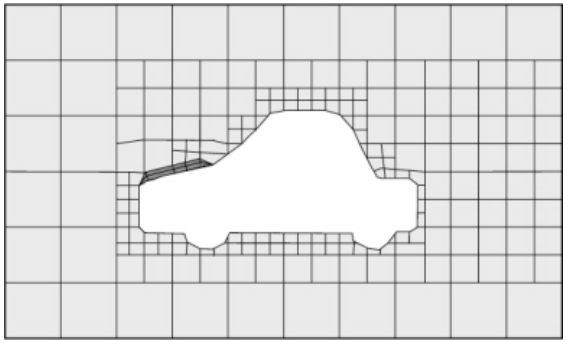
STL表面の細分割



階段状格子



境界適合



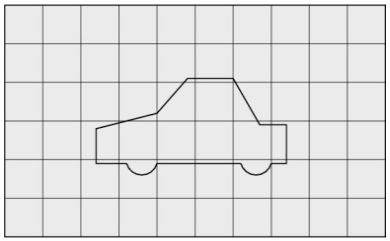
レイヤ付加

ベース六面体を元に複雑格子を生成

(図引用元: OpenFOAM User Guide Version 1.5)

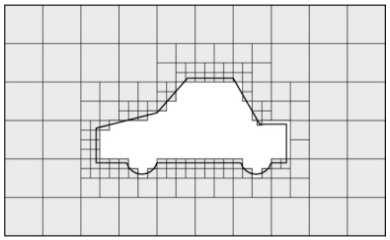
OpenCAE

格子の生成プロセス

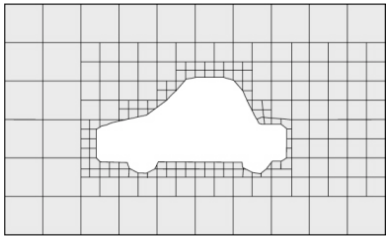


ベースの
六面体格子

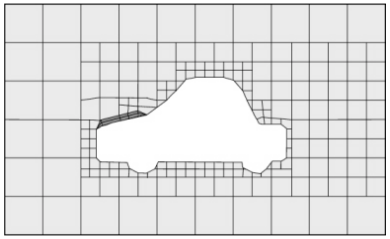
格子生成場所：
constant/polyMesh



階段状格子
1/polyMesh



境界適合
2/polyMesh



レイヤ付加
3/polyMesh

注)overwriteオプションを付けると
constant/polyMeshに上書き

blockMesh → snappyHexMesh

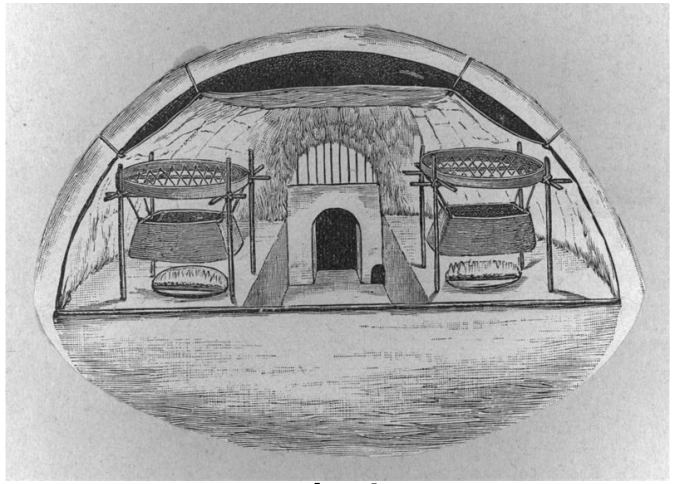
OpenCAE

snappyHexMeshの設定
(前編)

OpenCAE

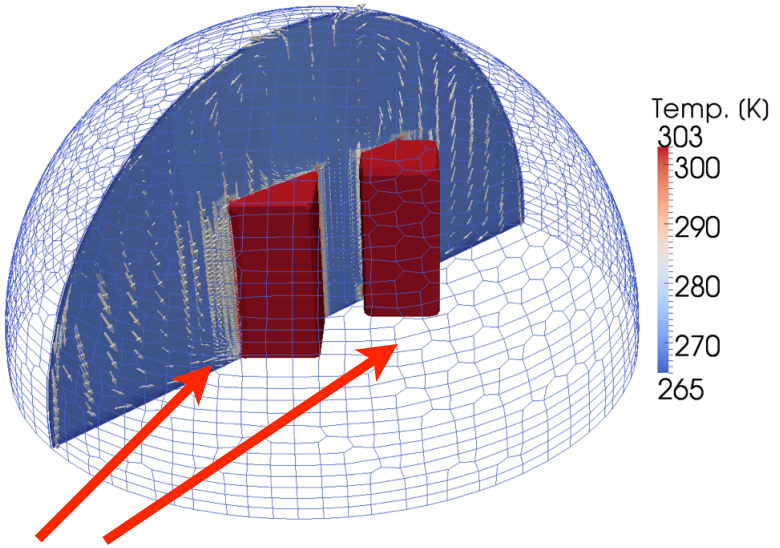
iglooWithFridgesチュートリアル

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

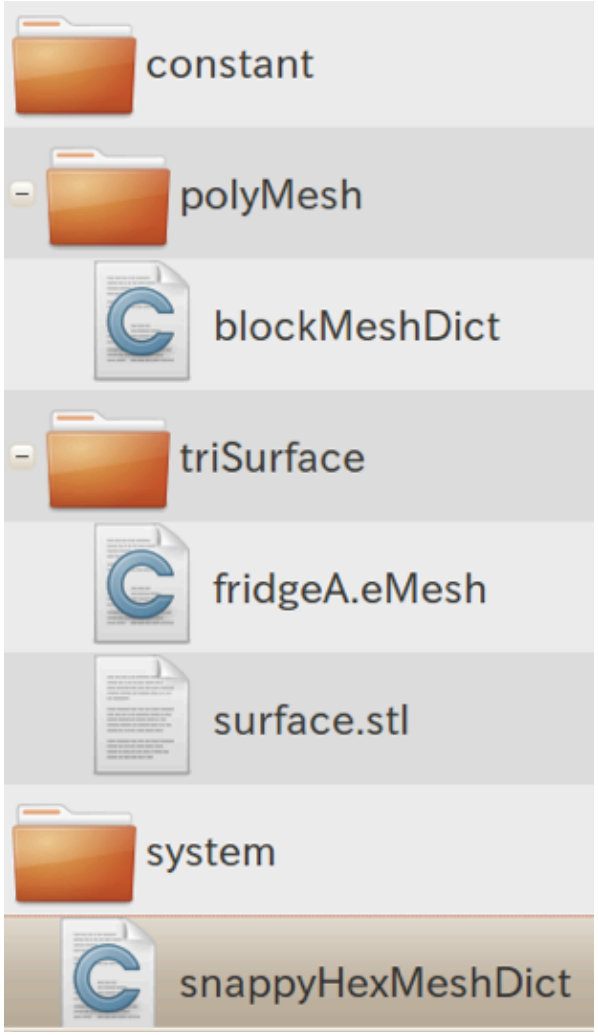


冷蔵庫?の発熱によるイグルー内の熱対流を解く

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

OpenCAE

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

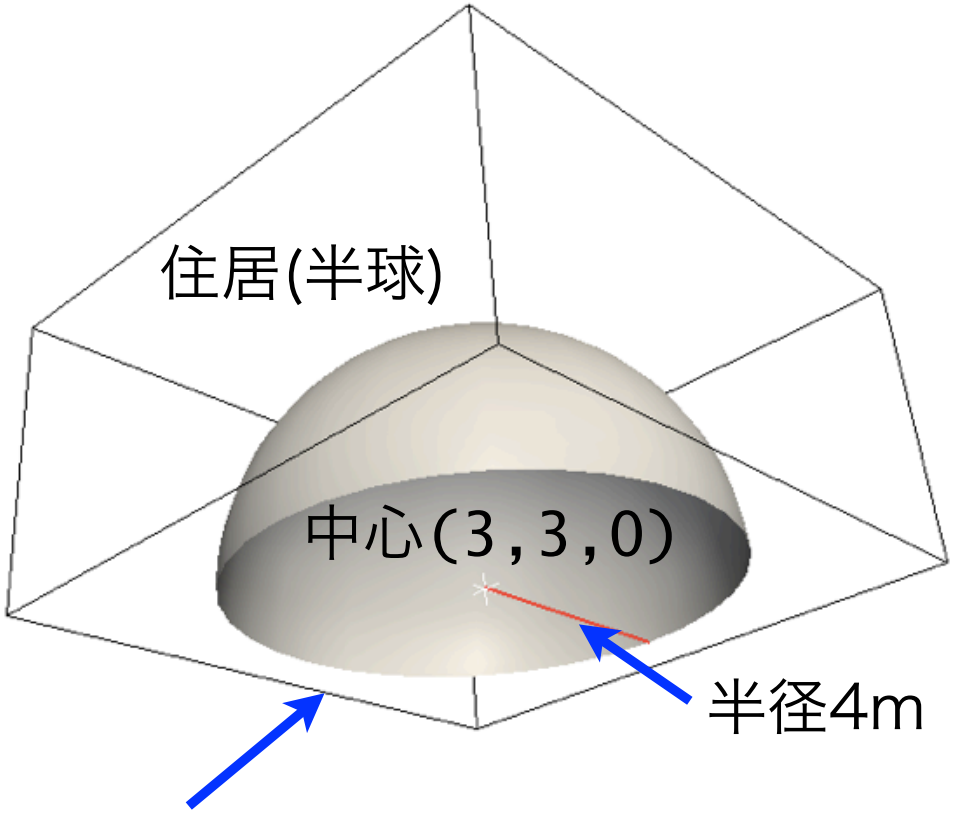
…表面・領域ベース細分割用のSTL

…snappyHexMeshの設定

OpenCAE

blockMeshDict 1

```
vertices
(
    (-2.03 -2.0  0)
    ( 8.03 -2.0  0)
    ( 8.03  8.0  0)
    (-2.03  8.0  0)
    (-2.03 -2.0  5)
    ( 8.03 -2.0  5)
    ( 8.03  8.0  5)
    (-2.03  8.0  5)
);

blocks
(
    hex (0 1 2 3 4 5 6 7)
    (20 20 20)
    simpleGrading (1 1 1)
);
```



ベース領域：住居を囲うよう設定

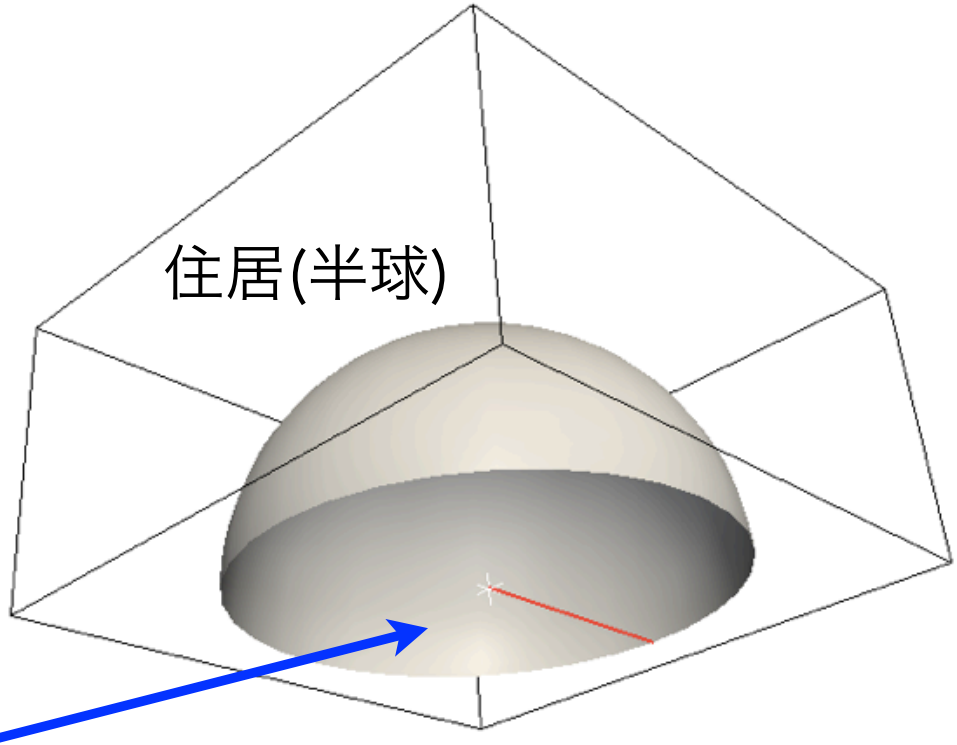
幅10m強×奥行10m×高さ5m

(なぜ幅が中途半端のは後述)

OpenCAE

blockMeshDict 2

```
patches
(
    wall ground
    (
        (0 3 2 1)
    )
);
```



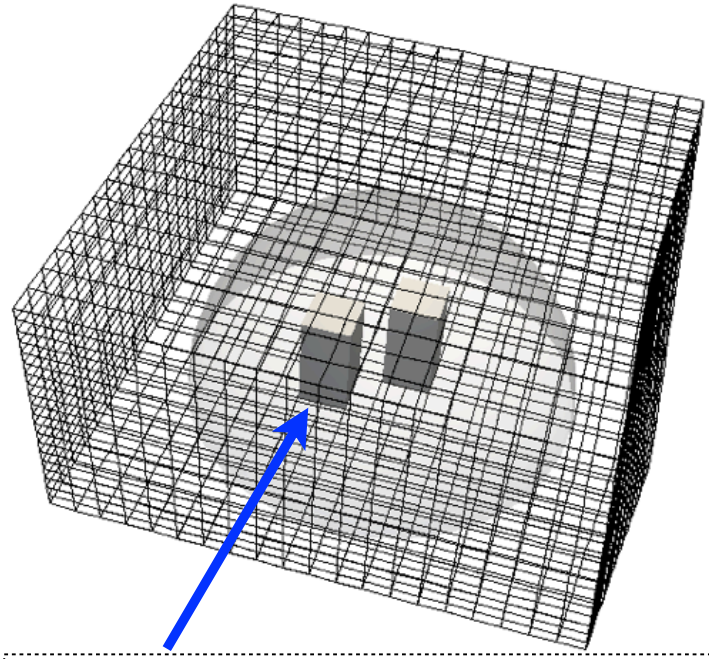
SHM後には地表面(ground)の境界面しか残らないので、この面のみパッチを定義

参考）残りの5面は自動的にdefaultFacesというパッチ名がつけられる

OpenCAE

ベース格子の条件

- ベース格子はすべて六面体格子(hex)である必要がある
- 格子のアスペクト比は概ね1にする。特に適合する表面付近で1から離れていると格子生成が遅くなったり失敗する
- 表面形状は少なくとも1つ格子と交差していないと、その表面は認識されない



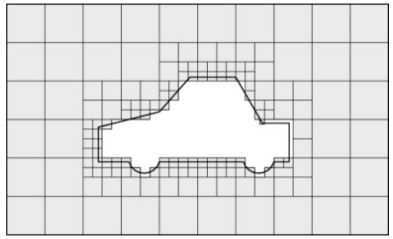
仮に冷蔵庫よりベース格子が大きくて、交差しない場合、冷蔵庫周りに格子が生成されない

OpenCAE

snappyHexMeshDict 1

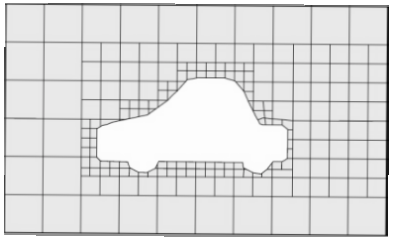
階段状格子

```
castellatedMesh true;
```



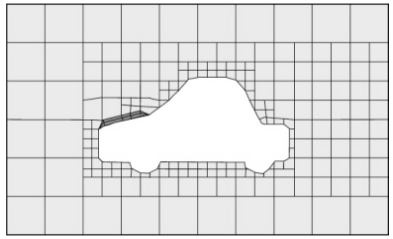
境界適合

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



レイヤ付加

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



OpenCAE

snappyHexMeshDict 2

geometry 各種形状の定義

castellatedMeshControls 階段状格子の制御設定

snapControls 境界適合の制御設定 (次回予定)

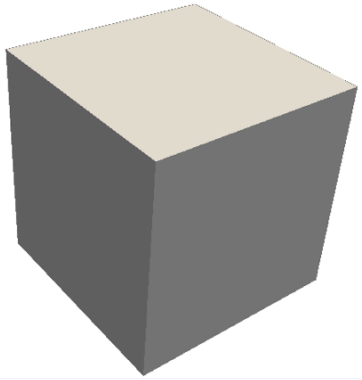
addLayersControls レイヤ付加の制御設定 (次回予定)

meshQualityControls 格子品質の制御設定 (次回予定)

OpenCAE

geometry 1

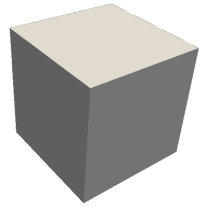
```
geometry
{
    igloo 名前
    {
        type searchableSphere; 球
        centre (3 3 0); 中心座標
        radius 4; 半径
    }
    box1 名前
    {
        type searchableBox; 立方体
        min (0 0 0); Boxの最小座標
        max (1 1 1); Boxの最大座標
    }
    fridgeFreezer 実際には使われないgeometry
```

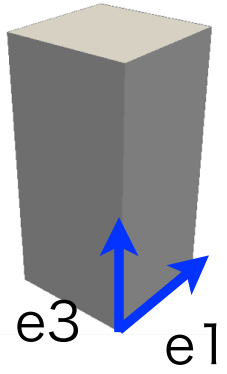


OpenCAE

geometry 2

```
twoFridgeFreezers 名前
{
    type searchableSurfaceCollection; 表面複合型
    mergeSubRegions true; パッチ名にregion名を付けるか
    seal 名前 (パッチ名はtwoFridgeFreezers_seal_0になる)
    {
        surface box1; 前に定義した高さ1の立方体
        scale (1.0 1.0 2.1); z方向に2.1倍
        transform 座標変換
        {
            type    cartesian; 座標系
            origin  (2 2 0); 移動ベクトル
            e1      (1 0 0); x軸方向の回転後ベクトル
            e3      (0 0 1); z軸方向の回転後ベクトル
        } この場合は回転しない
    }
```

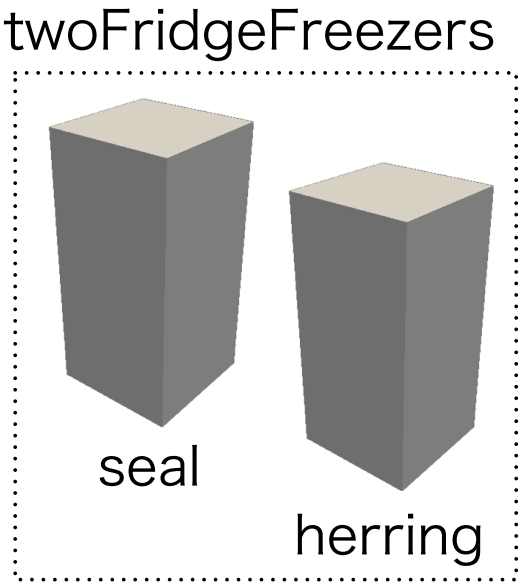




OpenCAE

geometry 3

```
twoFridgeFreezers
{
    seal
(続き)
    herring 名前 (パッチ名はtwoFridgeFreezers_herring_1になる)
     {
          surface box1; 高さ1の立方体
          scale (1.0 1.0 2.1); z方向に2.1倍
          transform 座標変換
          {
               type    cartesian;
               origin  (3.5 3 0); 移動ベクトル
(e1,e3は省略)
          }
     }
```



OpenCAE

geometry 4

他にも以下の型が指定できるが、詳しい使い方のドキュメントは無いのでソースを調べる必要がある

```
searchablePlane
searchablePlate
searchableSurfaceWithGaps
distributedTriSurfaceMesh
triSurfaceMesh
```

しかし複雑な形状についてはtriSurfaceMeshを使ってSTLファイルで形状を指定するのが一般的

OpenCAE

castellatedMeshControls 1

```
castellatedMeshControls 階段状格子の制御設定
{
    プロセッサあたりの格子最大数
    maxLocalCells 1000000;

    全体の格子最大数
    maxGlobalCells 2000000;
    この値を超えると細分割を行わない。
    表面形状内の格子除去前にもこの制限が適用される。

    細分割される格子最小数
    minRefinementCells 100;
    細分割候補数がこれ未満になると細分割を止める。
```

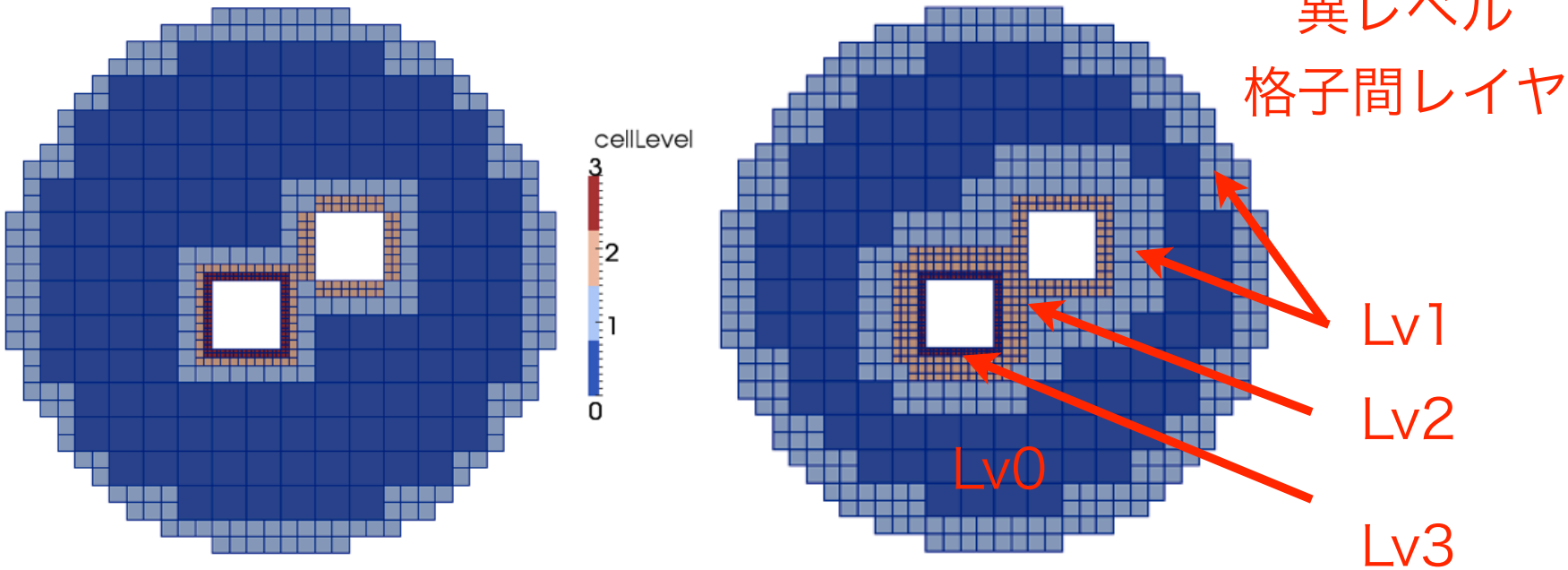
OpenCAE

castellatedMeshControls 2

細分割レベルが異なる格子間のレイヤ数

```
nCellsBetweenLevels 1;
```

大きい値になると滑らかな細分割になり格子数も増える



nCellsBetweenLevels 1　　nCellsBetweenLevels 2

groundパッチの格子レベル

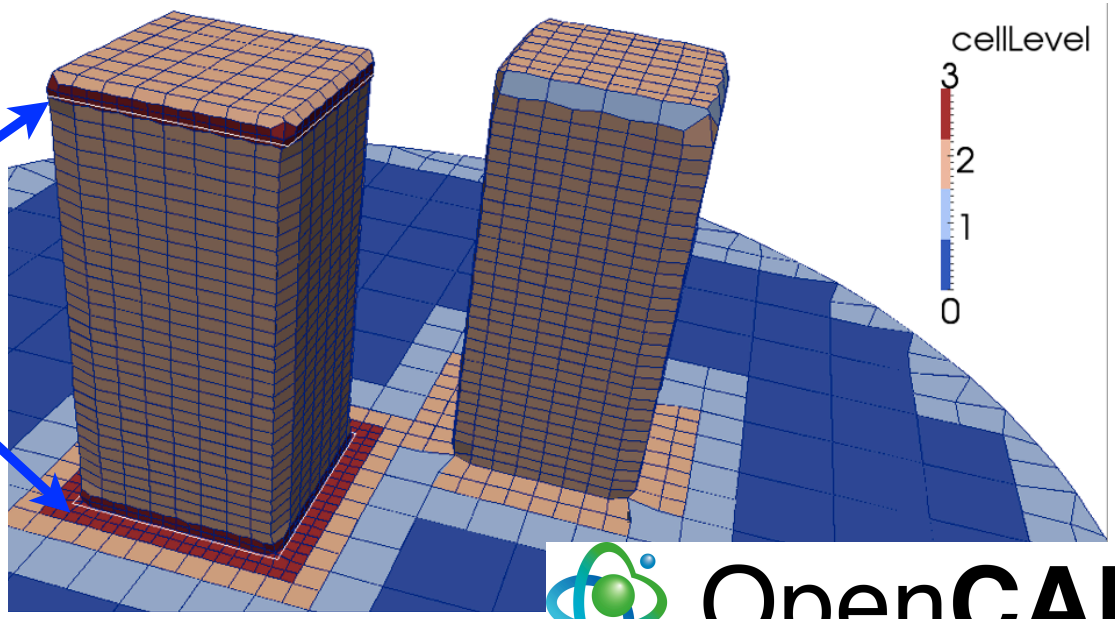
OpenCAE

castellatedMeshControls 3

特徴線と交差する格子が指定されたレベルに細分割される

```
features 特徴線
(
  {
    file "fridgeA.eMesh"; 特徴線のファイル
    level 3; 細分割レベル。triSurfaceディレクトリに置く
  }
);
```

特徴線(白線)
これと交差する
格子がLv3分割

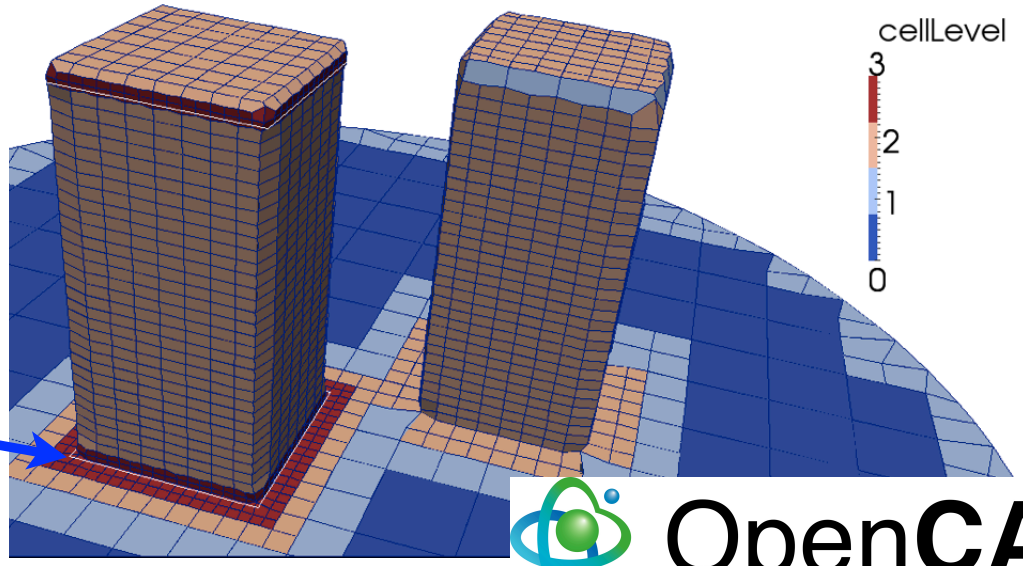


OpenCAE

castellatedMeshControls 4

constant/triSurface/fridgeA.eMesh:

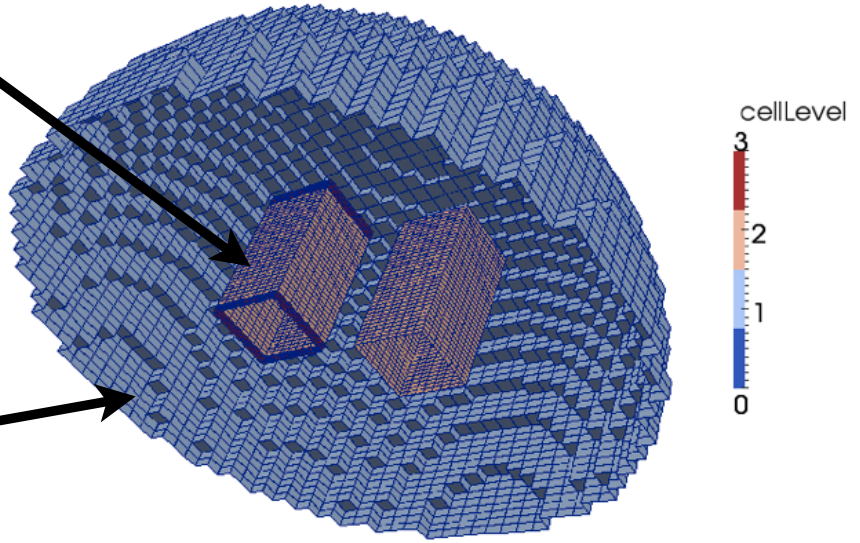
```
// Points 節点座標の定義

(
(1.99 1.99 0.01) //0
(3.01 1.99 0.01) //1
(3.01 3.01 0.01) //2
(1.99 3.01 0.01) //3
)

// Edges 辺の定義

(
(0 1)
(1 2)
(2 3)
(3 0)
)
```



OpenCAE

castellatedMeshControls 5

```
refinementSurfaces 表面ベースの細分割
{
  パッチ名(blockMeshのパッチ名やgeomety名)
  twoFridgeFreezers
  {
    表面ベースの細分割(最小レベル、最大レベル)
    level (2 2); レベル2
  }
  パッチ名(“正規表現も可”)
  "iglo.*" igloで始まる名前
  {
    level (1 1); レベル1
  }
}
```



OpenCAE

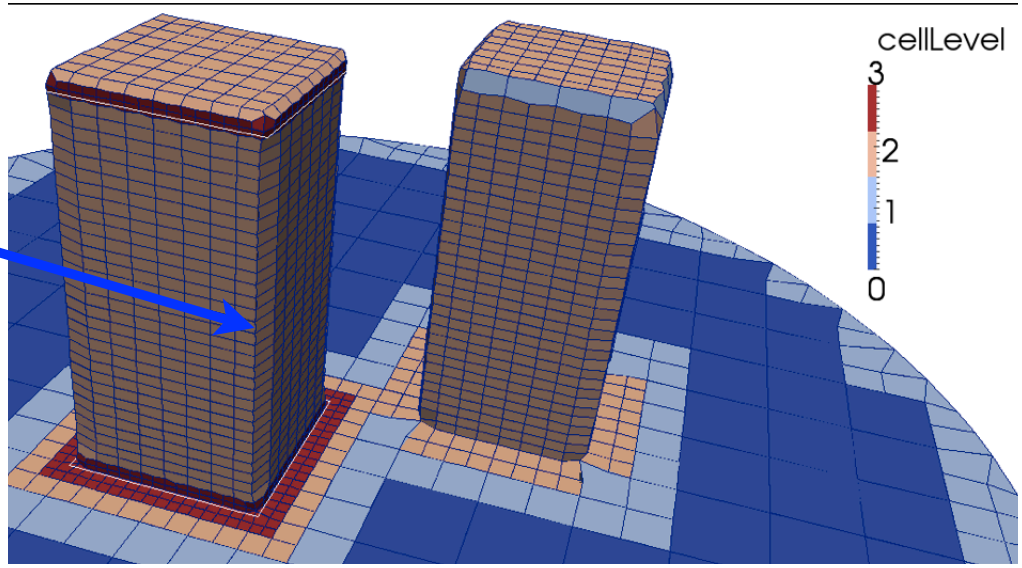
castellatedMeshControls 6

特徴線と見なす面の交差角度

```
resolveFeatureAngle 60;
```

この角度を超えて面が交わっている辺は特徴線と見なし、その辺の周辺の格子は最大レベルまで細分割する

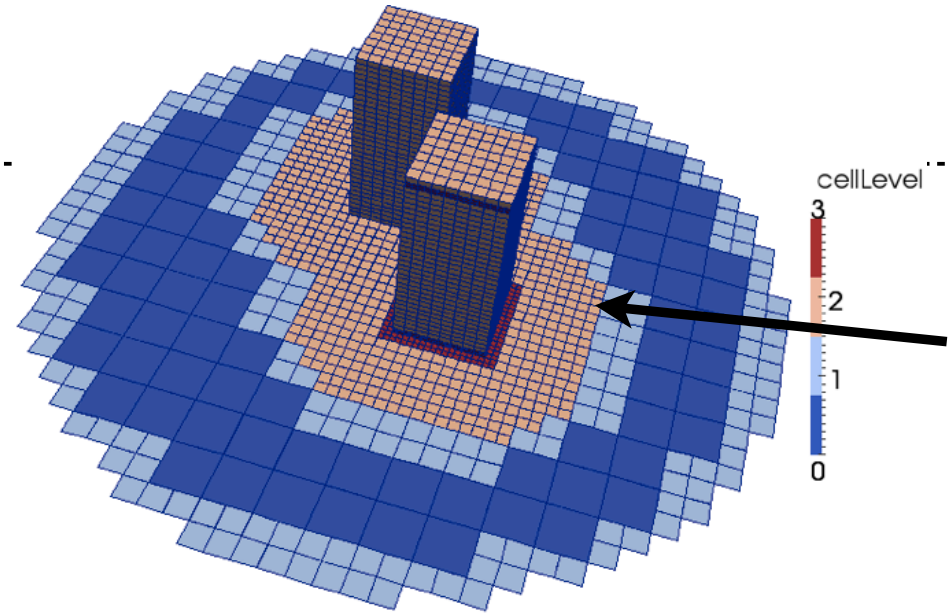
60度以上で交差している角は特徴線とみなされるが、ここでは最大レベルと最小が同じなので細分割されない



OpenCAE

castellatedMeshControls 7

```
refinementRegions 領域ベース細分割の領域定義
{ 注)以下の定義は元のチュートルアルにはなく付け加えたもの
  twoFridgeFreezers geometry名
  {
    mode distance; 領域からの距離が指定値内の格子が細分割
    levels ((1.0 2)); (距離 細分割レベル) 複数指定可
  }
}
```

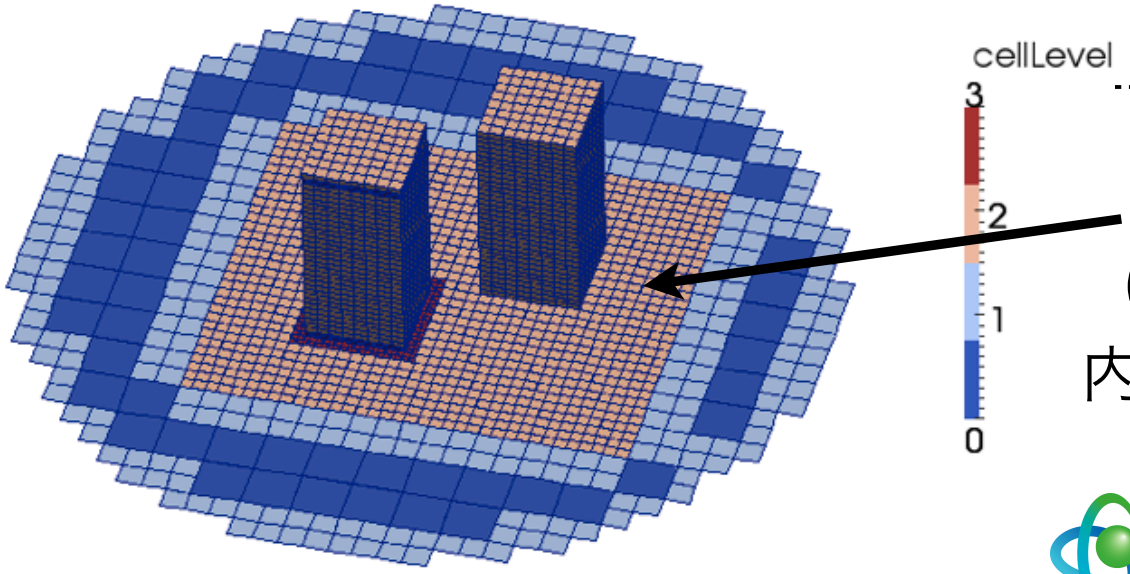


冷蔵庫からの距離が
1.0以内の格子がレベル
2に細分割された

OpenCAE

castellatedMeshControls 8

```
refinementRegions 領域ベース細分割の領域定義
{ 注)以下の定義は元のチュートルアルにはなく付け加えたもの
   box2 {
      mode inside; 領域内の格子が細分割(outside もある)
      levels ((1.0 2)); (ダミーの値 細分割レベル)
   }
}
```



領域box2
(後の演習で定義)
内部の格子が細分割

OpenCAE

castellatedMeshControls 9

格子内部点

```
locationInMesh (3 0.28 0.43);
```

この点を元にgeometyの内外判定がなされる。

この点は、ベース格子のみならず、細分割された格子においても格子の界面上にあってはならず、必ず格子の内側にあることに注意。

```
allowFreeStandingZoneFaces true;
```
次回の演習で説明予定

注)x座標が3なので界面上にありそうだが、

x座標の範囲は-2.03から8.03と**きり**が良くないので

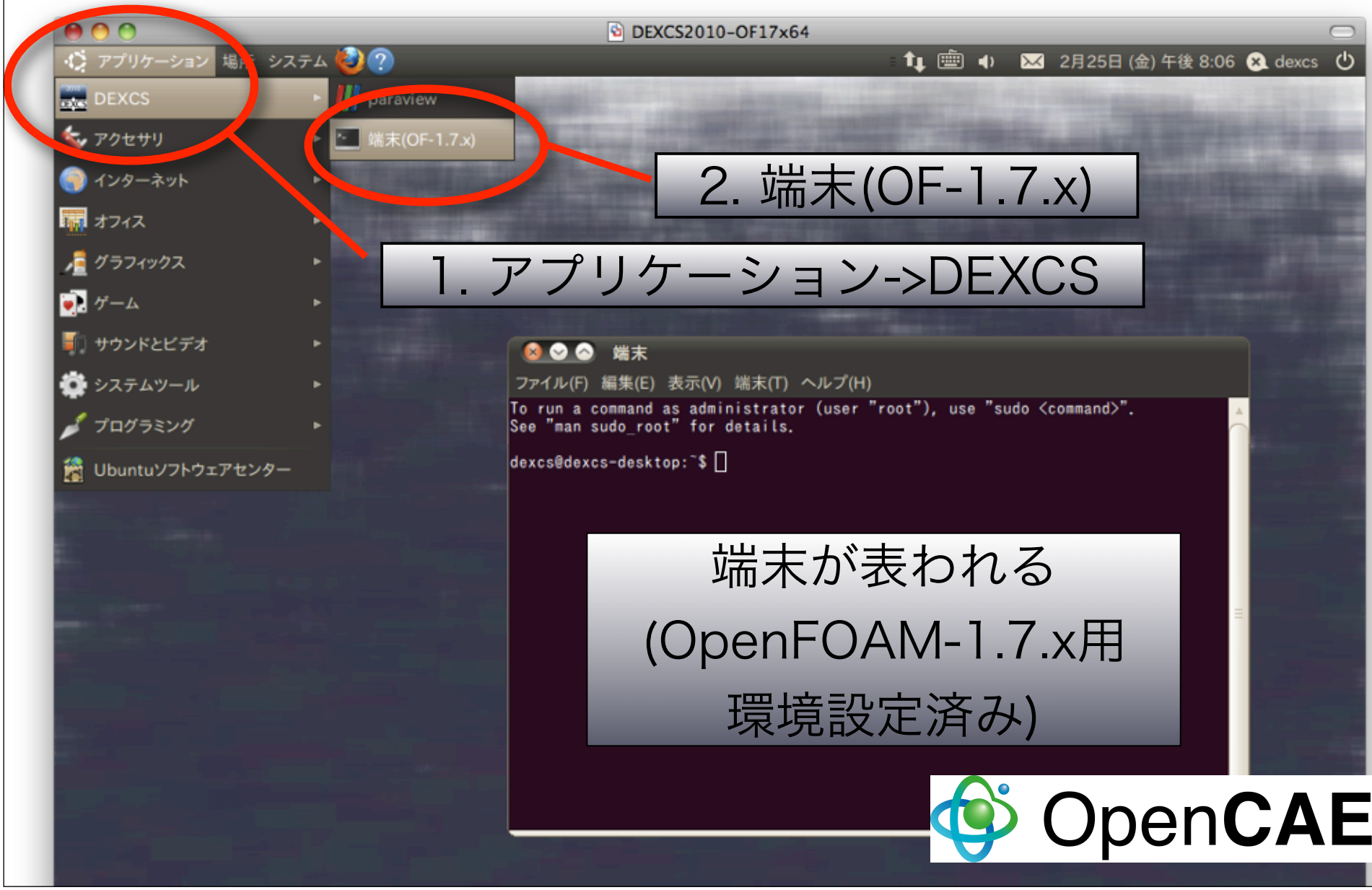
界面上にはなることはない。

OpenCAE

snappyHexMeshの演習
(前編)

OpenCAE

端末の起動


DEXCS2010-OF17x64
アプリケーション
場所
システム
2月25日 (金) 午後 8:06
dexcs
DEXCS
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
Ubuntuソフトウェアセンター
paraview
端末(OF-1.7.x)
2. 端末(OF-1.7.x)
1. アプリケーション->DEXCS
端末
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:~$
端末が表われる
(OpenFOAM-1.7.x用
環境設定済み)
OpenCAE

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 &↵
```

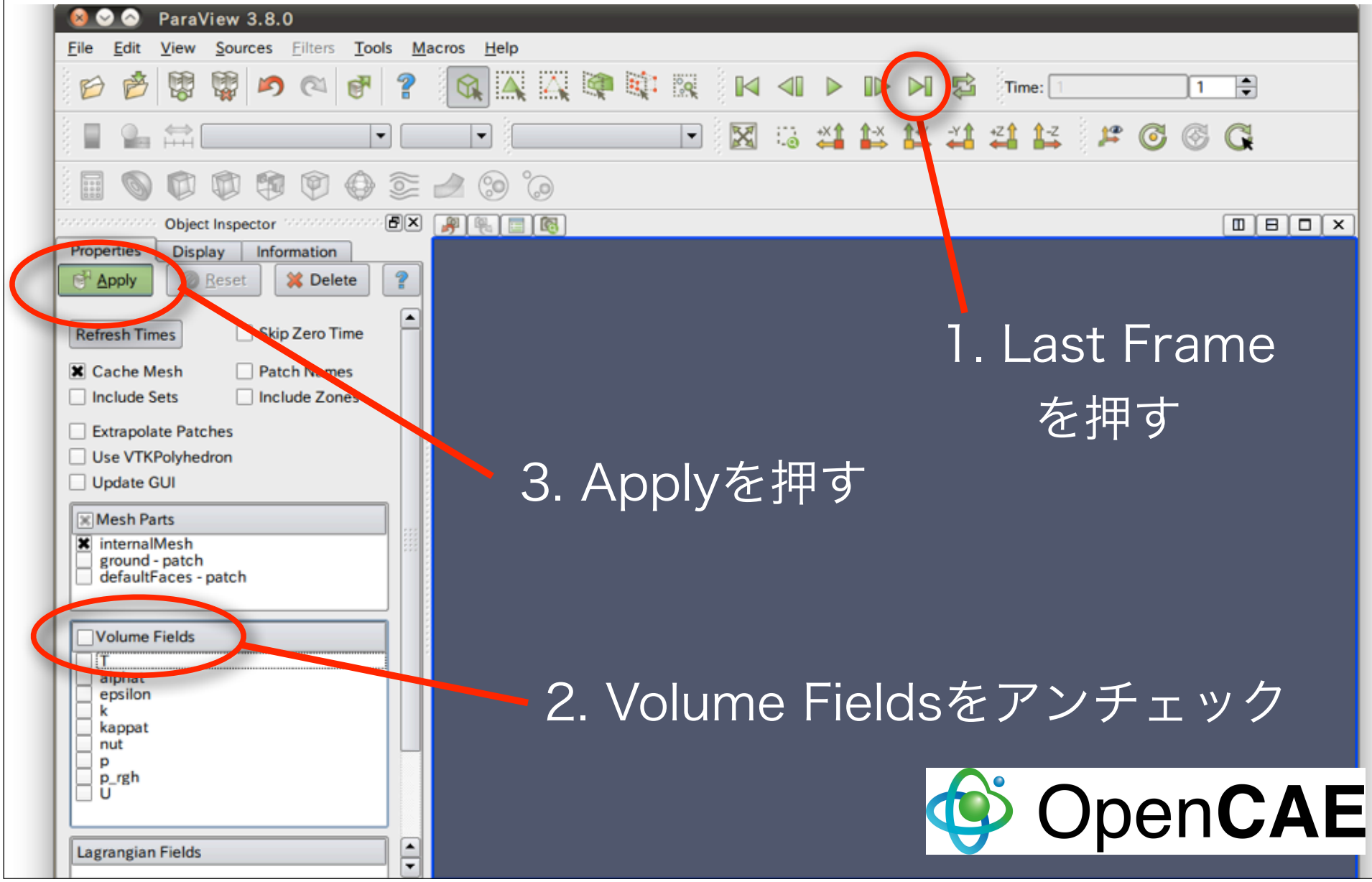
```
castellatedMesh true;  階段状格子
snap            false; ←変更 (境界適合しない)
addLayers       false; ←変更 (レイヤ付加しない)
```

格子生成とsnappyHexMeshのログの確認

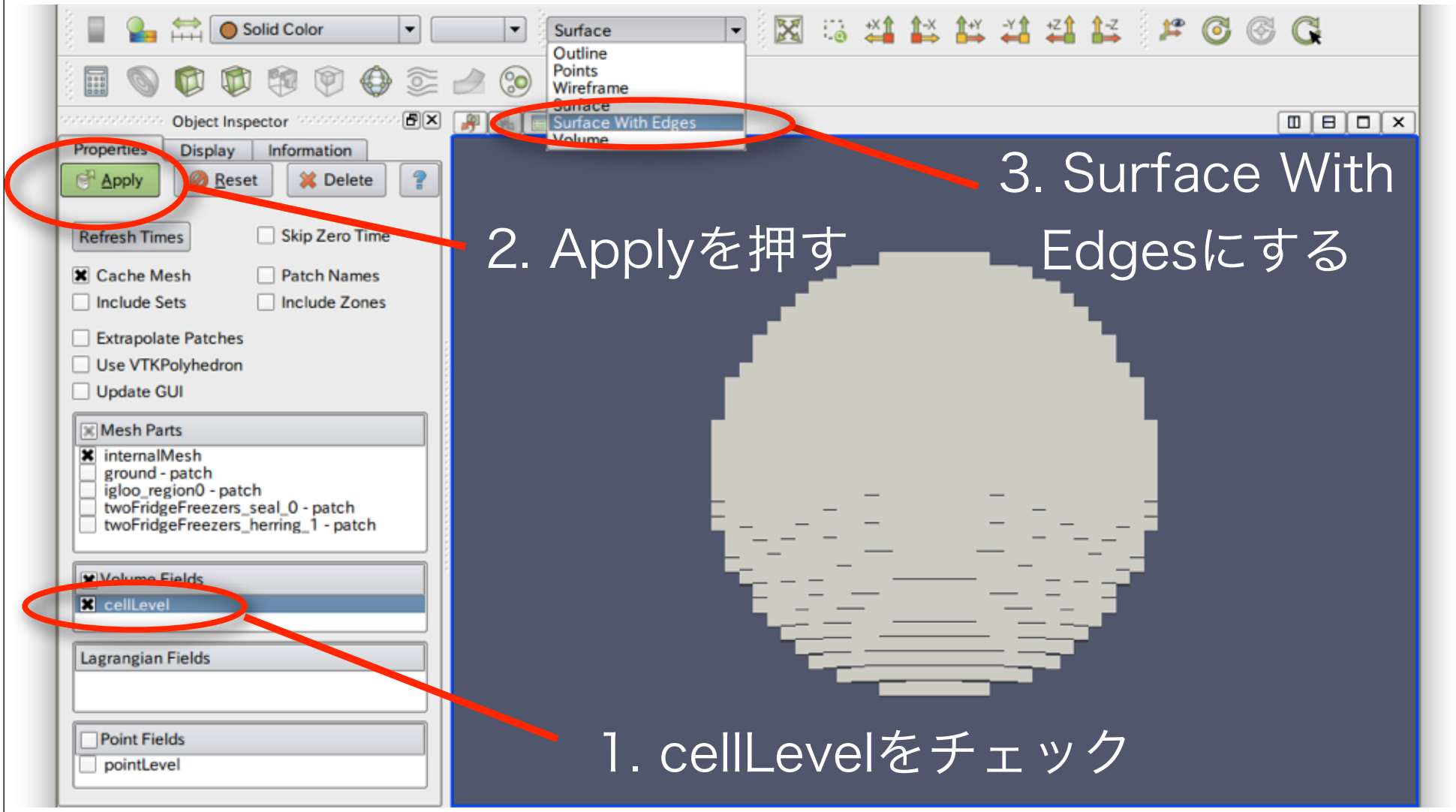
```
blockMesh↵ ベース格子生成
foamJob -s snappyHexMesh↵  ログを見ながら記録
more log↵ ログの確認
```

OpenCAE

paraFoamによる格子可視化 1

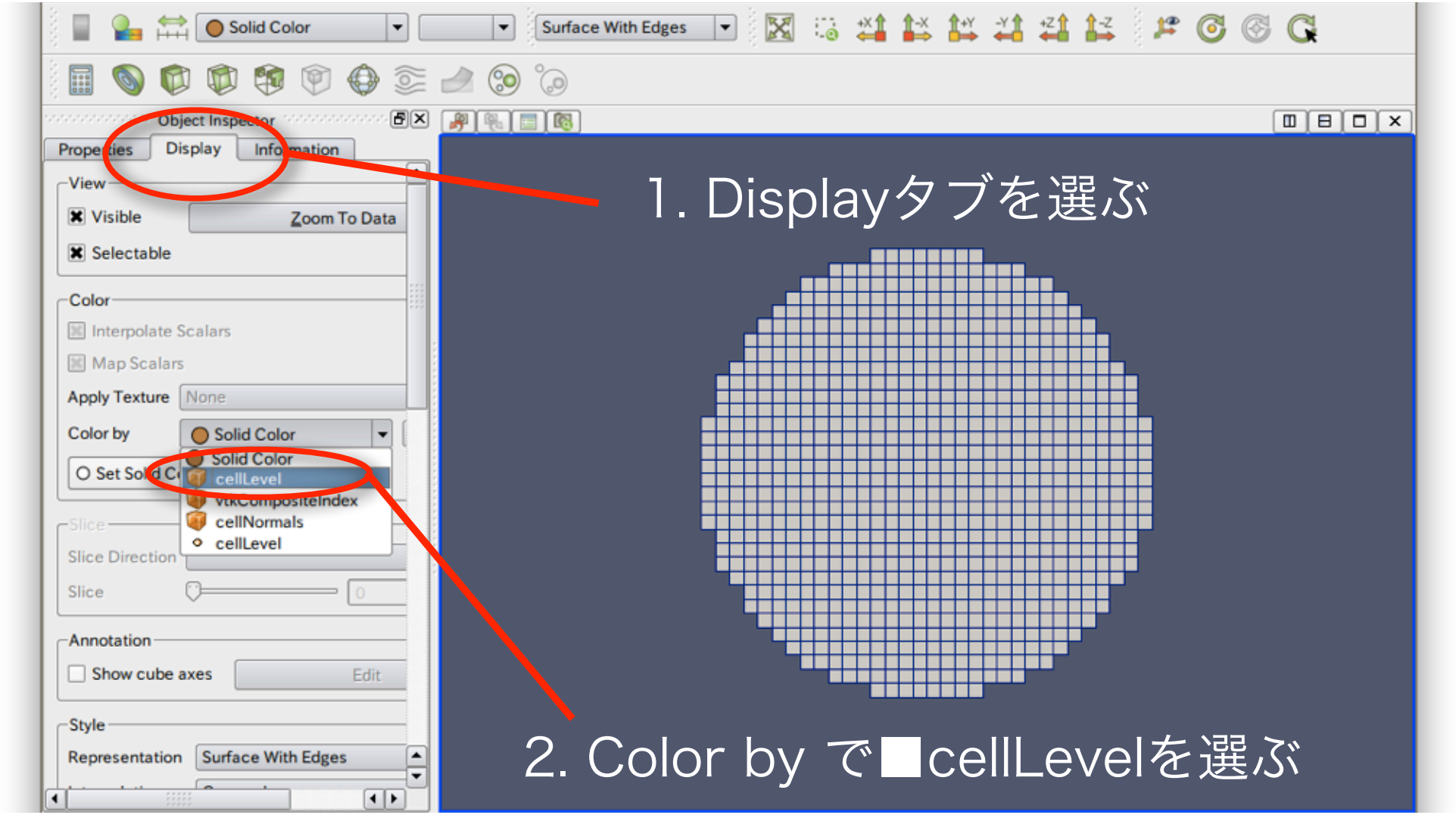


paraFoamによる格子可視化 2



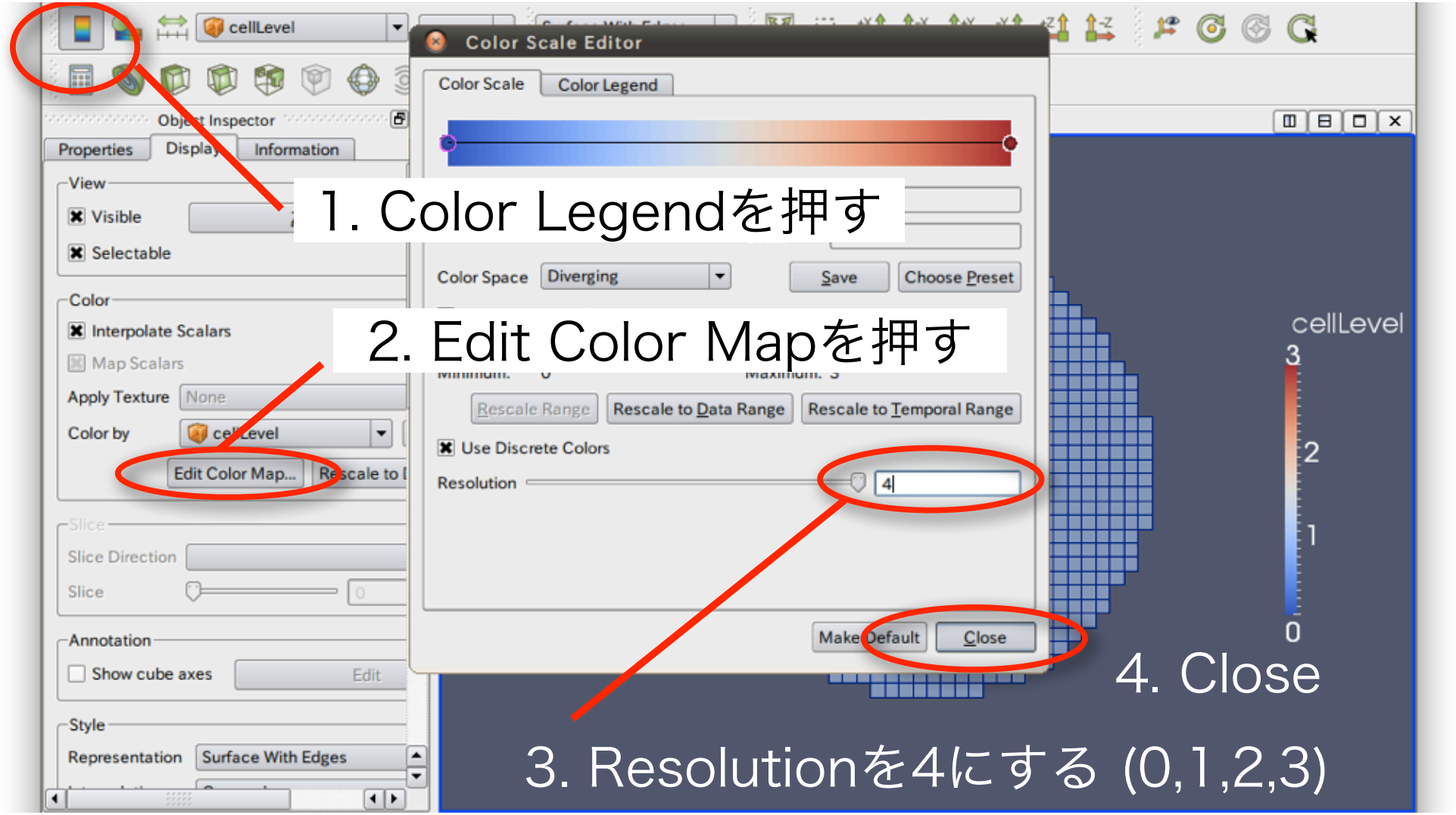
OpenCAE

paraFoamによる格子可視化 3



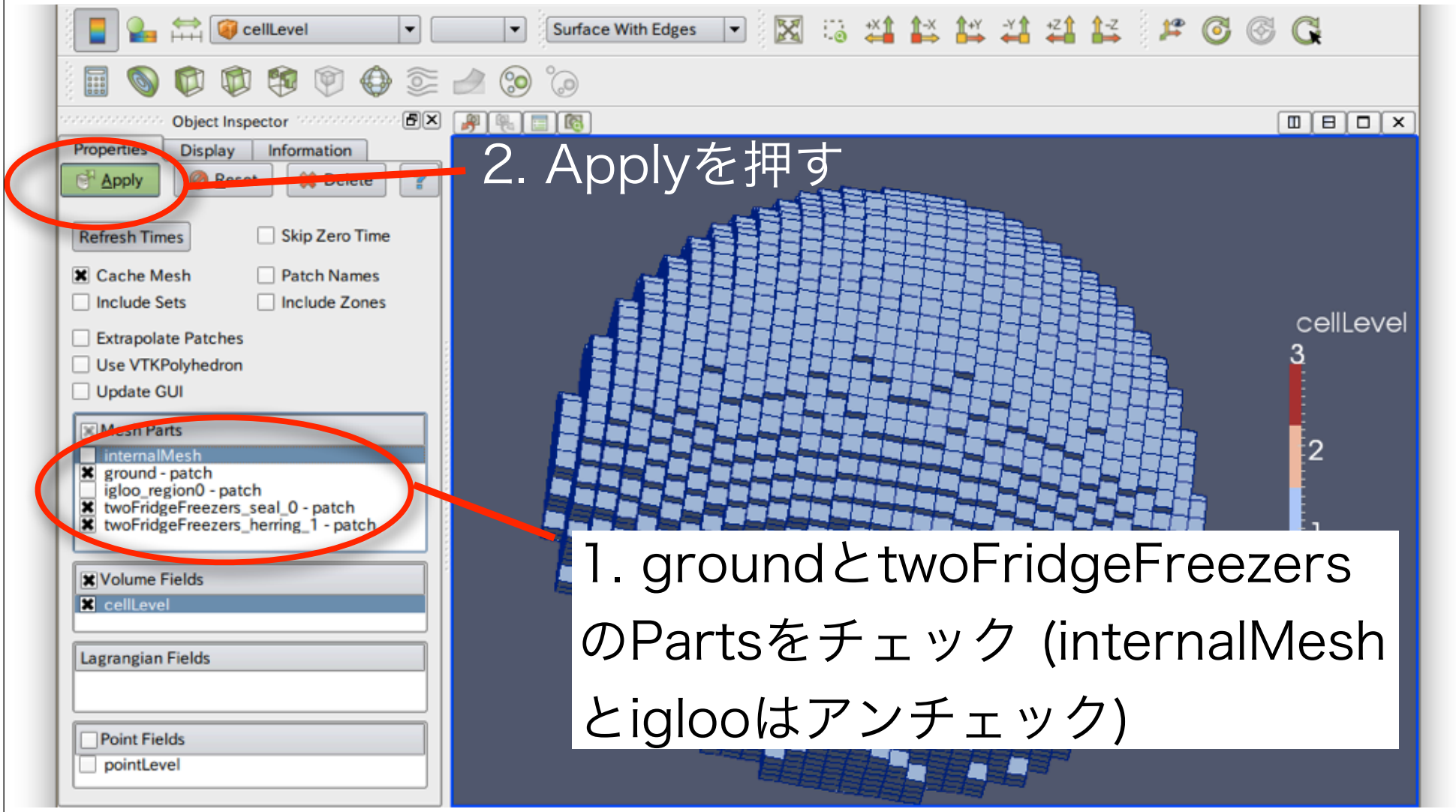
OpenCAE

paraFoamによる格子可視化 4



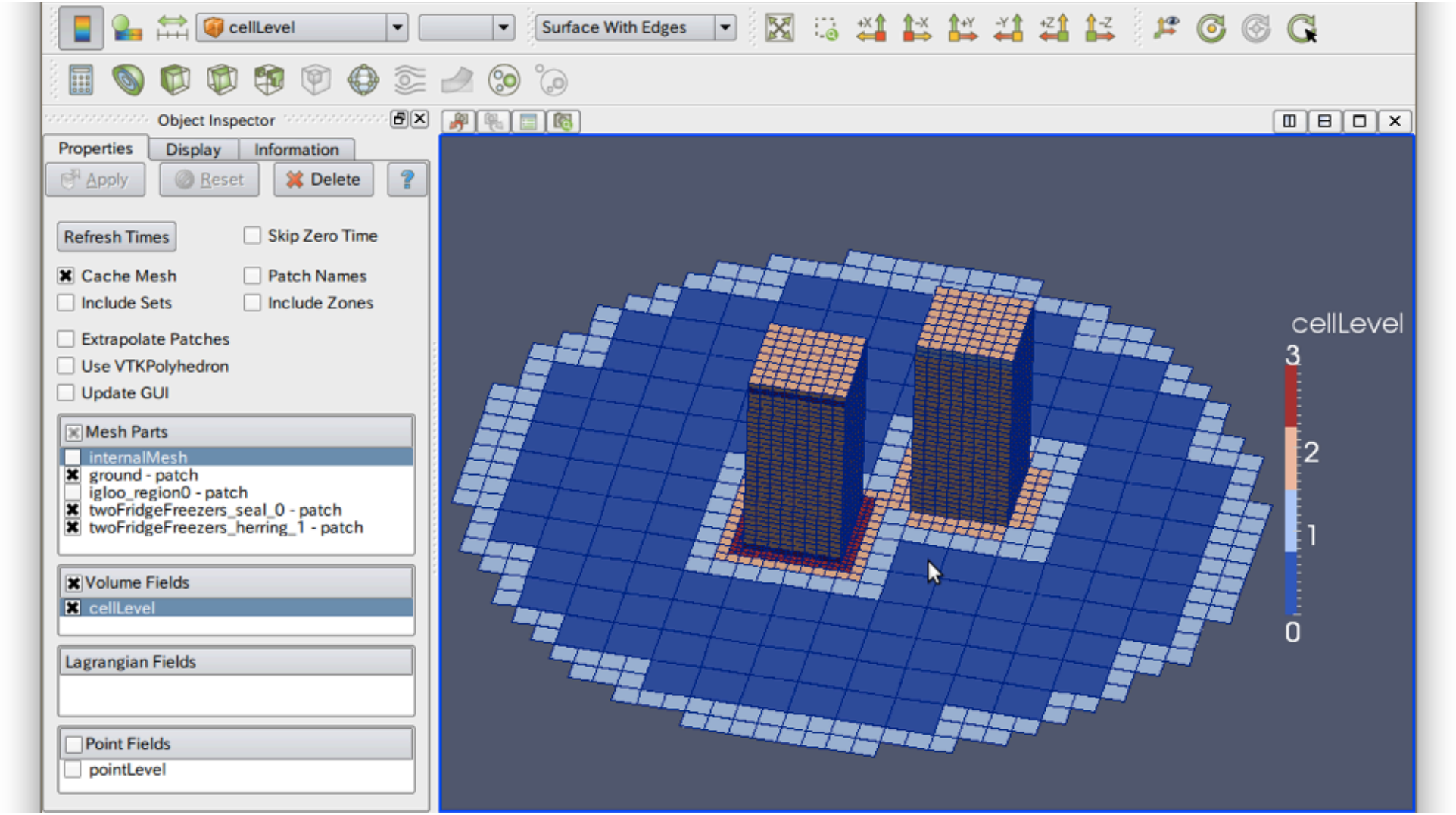
OpenCAE

paraFoamによる格子可視化 5



OpenCAE

paraFoamによる格子可視化 6



OpenCAE

演習1(表面レベル修正) 1

snappyHexMeshDictの修正

```
refinementSurfaces
{
:
  "iglo.*"
  {
    // Surface-wise min and max refinement level
    level (2 2); ←レベル2に変更
```

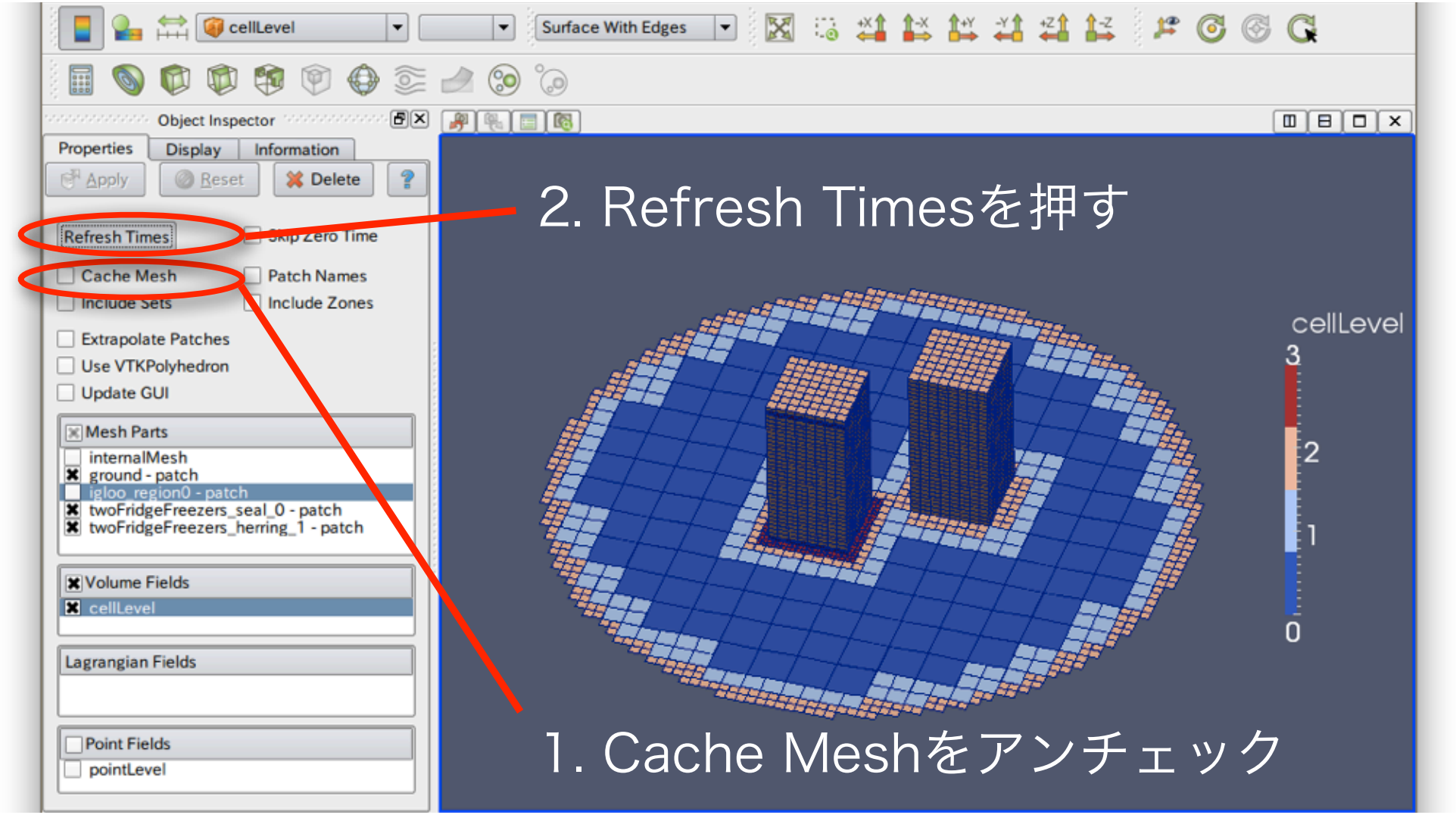
snappyHexMeshの再実行

↑ 前に行なったコマンドはカーソル↑で呼び出せる

```
foamJob -s snappyHexMesh⏎
```

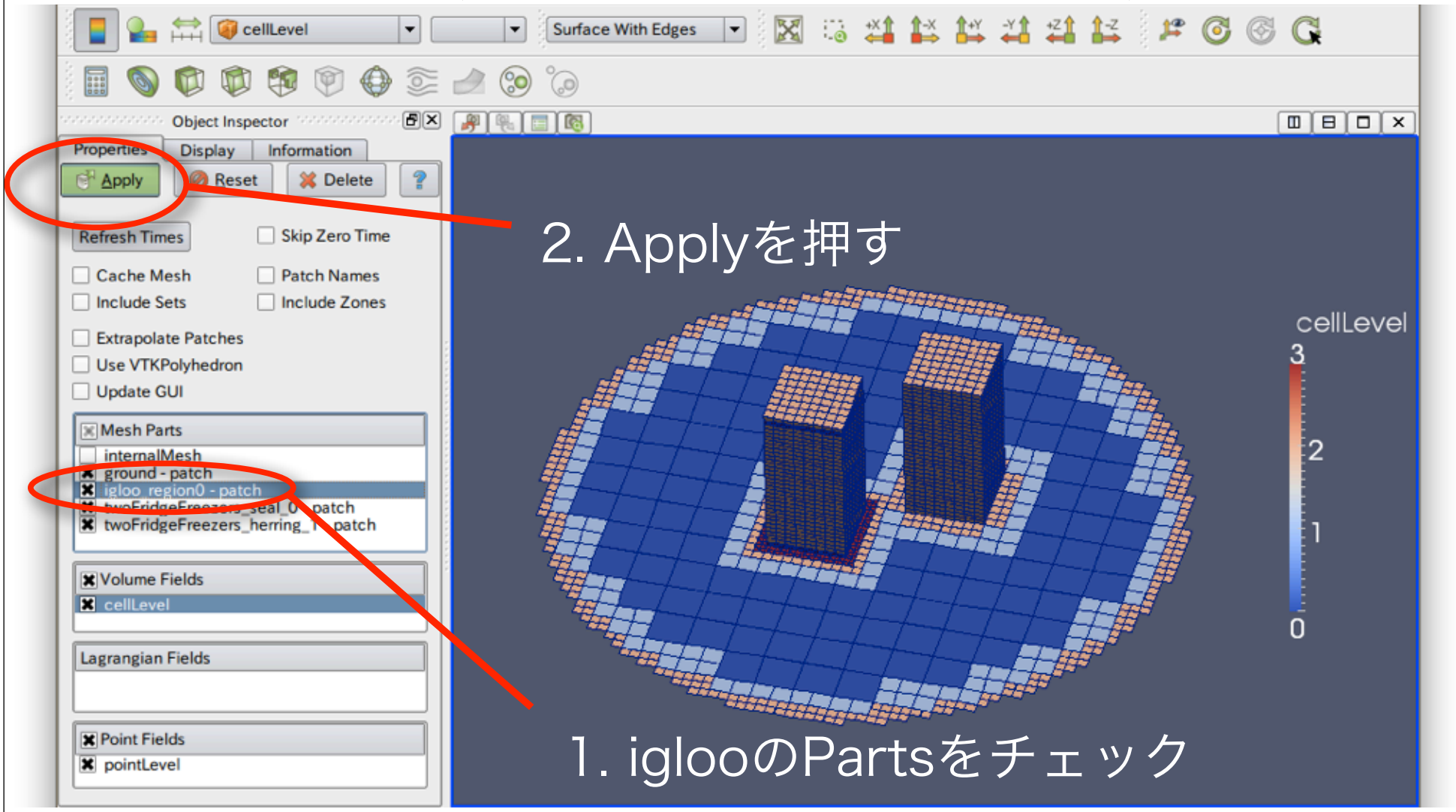
OpenCAE

演習1(表面レベル修正) 2



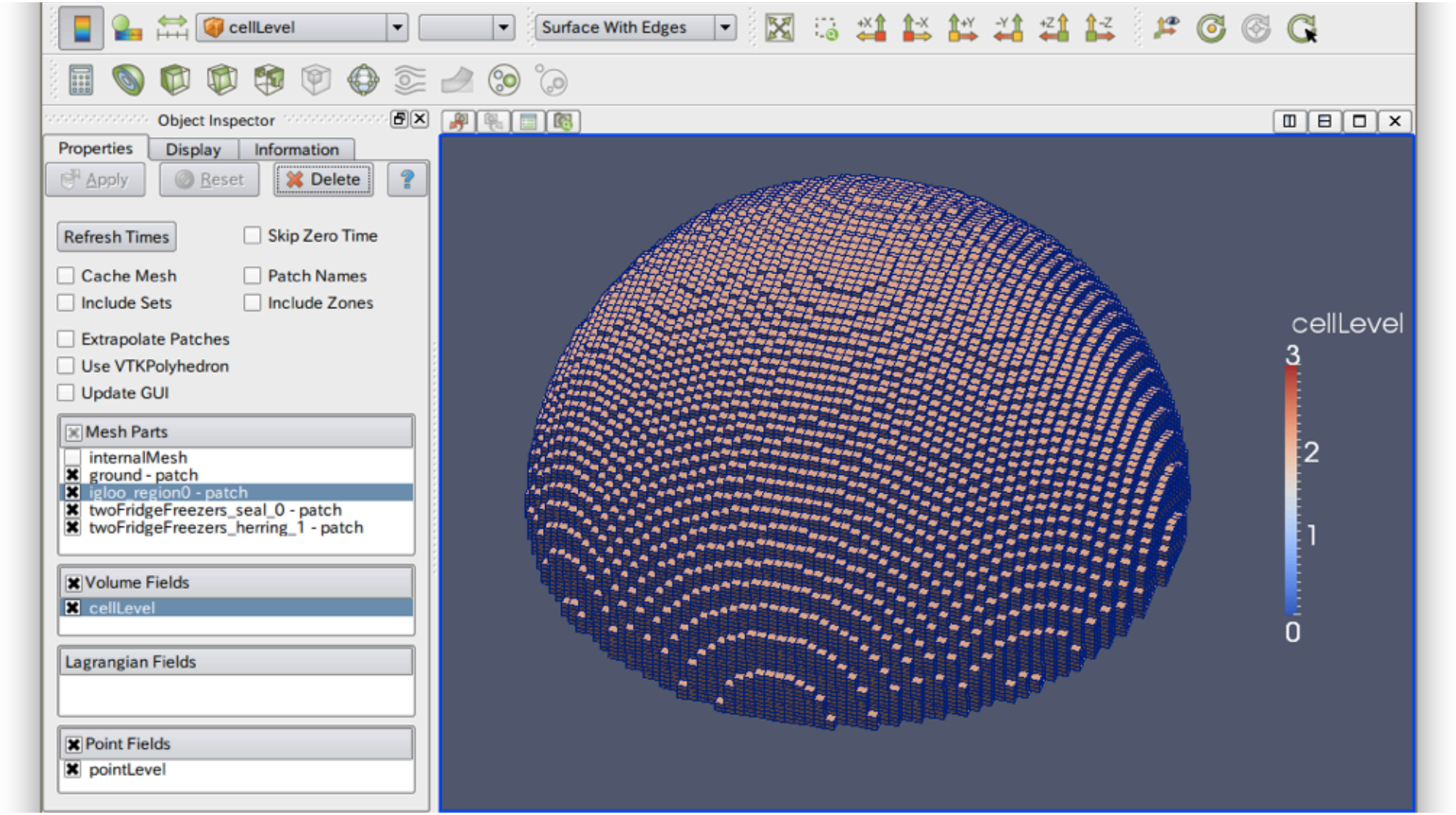
OpenCAE

演習1(表面レベル修正) 3



OpenCAE

演習1(表面レベル修正) 4



OpenCAE

演習2(領域ベース細分割A) 1

snappyHexMeshDictでigloo表面のレベルを1に戻す

```
"iglo.*"
{
    // Surface-wise min and max refinement level
    level (1 1); ←レベル1に戻す
```

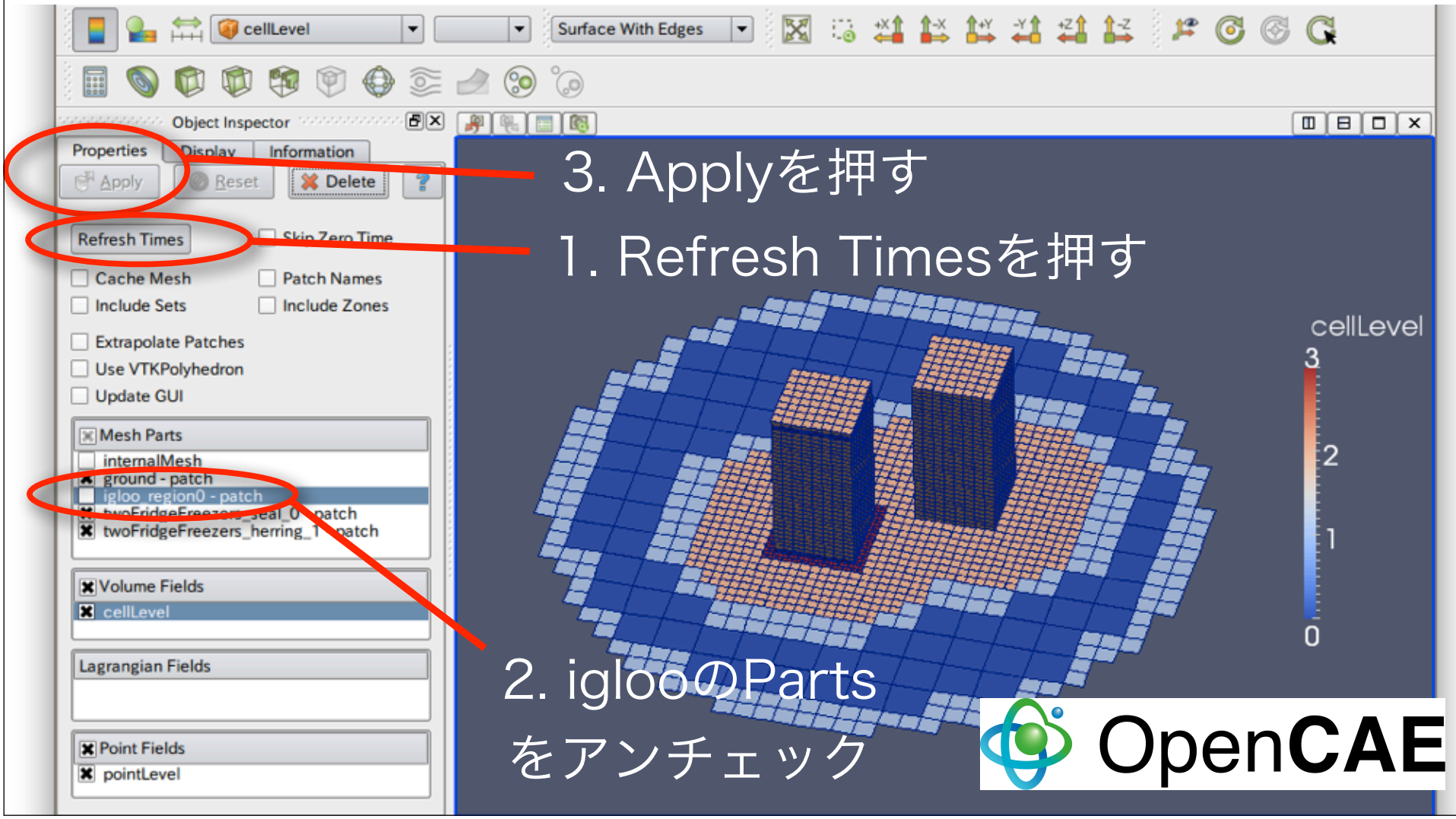
refinementRegionsに以下の赤字の行を足す

```
refinementRegions
{
    twoFridgeFreezers
    {
        mode distance; ←行末のセミコロンを忘れずに

        levels ((1.0 2)); ←levelsの”s”、括弧が二重に注意

    }
}
```

OpenCAE

演習2(領域ベース細分割A) 1

foamJob -s snappyHexMesh↵



演習2(領域ベース細分割B) 1

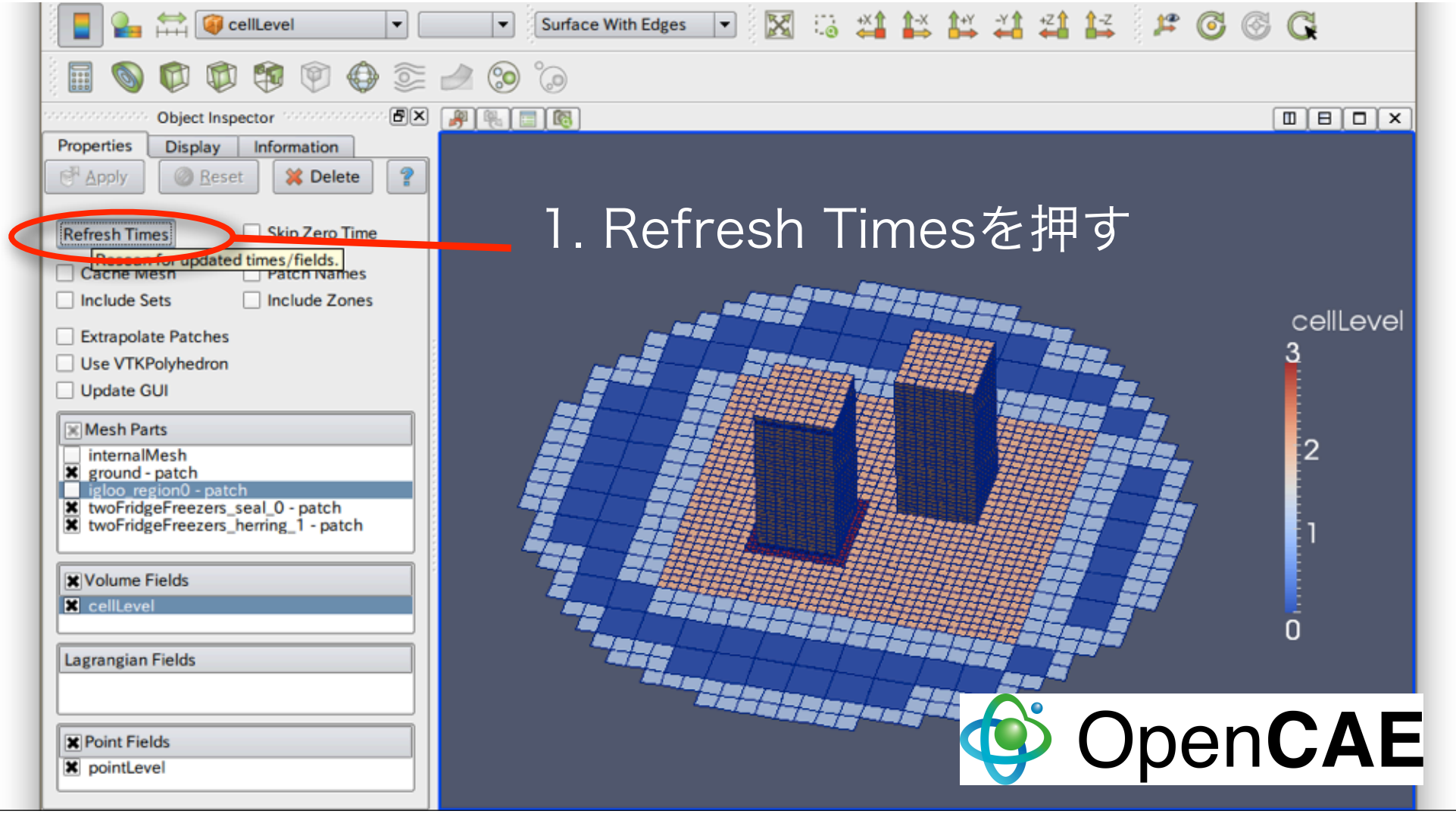
snappyHexMeshDictで細分割用領域box2を追加

```
geometry
{
    box2 ←box1の定義をコピーして書き換える
    {
        type searchableBox;
        min (1 1 0);
        max (5.5 5 3);
    }
(中略)
    refinementRegions
    {
        box2
        {
            mode inside;
            levels ((1.0 2));
         }
    }
```

OpenCAE

演習2(領域ベース細分割B) 2

foamJob -s snappyHexMesh↵



目次

1. snappyHexMeshの特徴
2. snappyHexMeshの設定(前編)
3. snappyHexMeshの演習(前編)
4. 質疑

OpenCAE