snappyHexMesh例題 : iglooWithFridges

2013/04/13

オープンCAE勉強会＠富山

中川慎二

snappyHexMesh を使う例題

- snappyHexMesh を使う例題が、OpenFOAMに含まれている。
  - ユーザーディレクトリ/OpenFOAM/user-2.1.1/run/tutorials/mesh/snappyHexMesh

STLを使わない方法

- 角柱、円柱、平面などの単純な形状は、STLファイルを用いることなく、snappyHexMeshDict 内に記述することで、使用することができます。
- geometry 内に、記述します。
- その記述方法については、「iglooWithFridges」チュートリアルのsnappyHexMeshDict が参考になります。

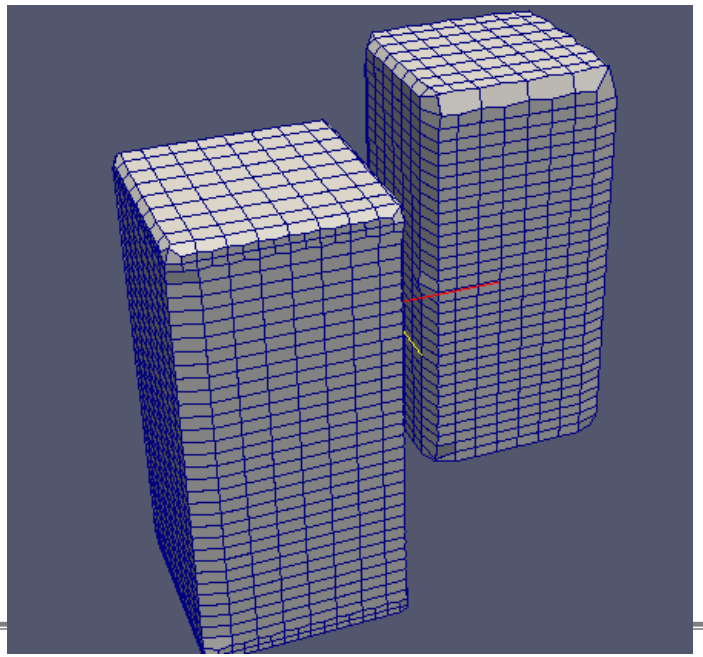
- snappyHexMeshDict で使えるsearchableSurface
  - closedTriSurfaceMesh
  - distributedTriSurfaceMesh
  - **searchableBox**
  - searchableCylinder
  - searchablePlane
  - searchablePlate
  - **searchableSphere**
  - **searchableSurfaceCollection**
  - searchableSurfaceWithGaps
  - triSurfaceMesh

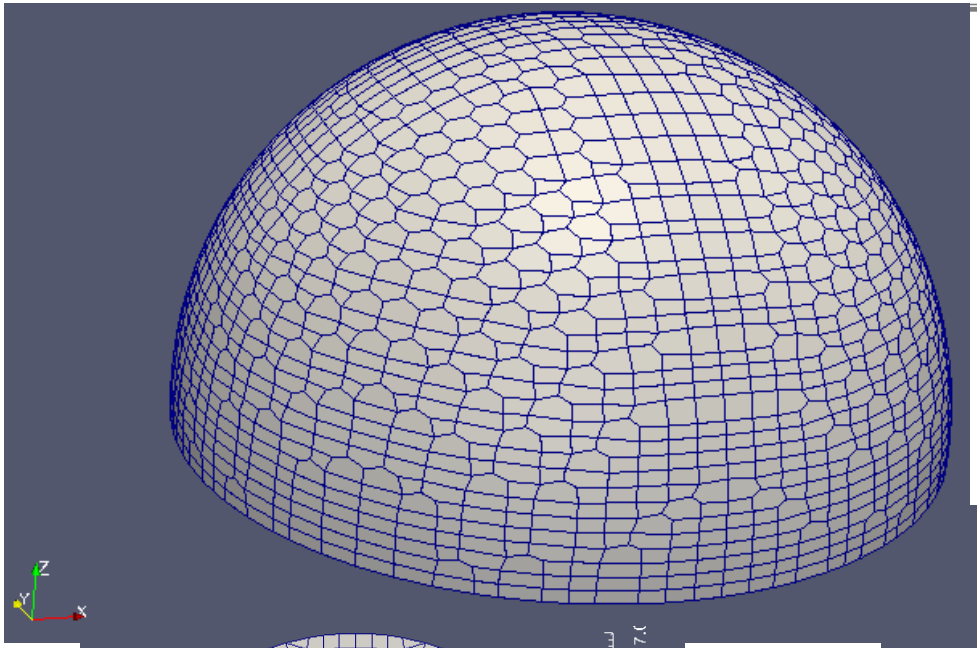
経緯

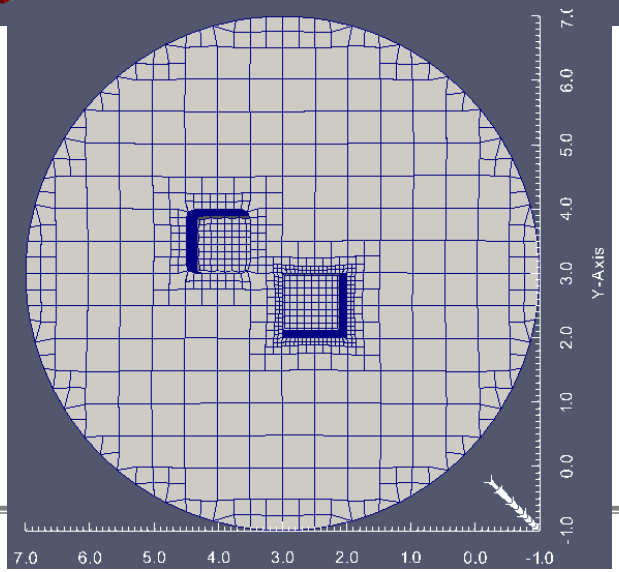
- 前回のsnappyHexMeshミニ講習会では、STLファイルで読み込んだ形状に合わせて、セルを生成した。
- 基本的な形状は、STLファイルを使わずに、snappyHexMeshDict内で寸法を指定して生成することができる。
- その方法については、iglooWithFridges例題を参考にして知ることができる。

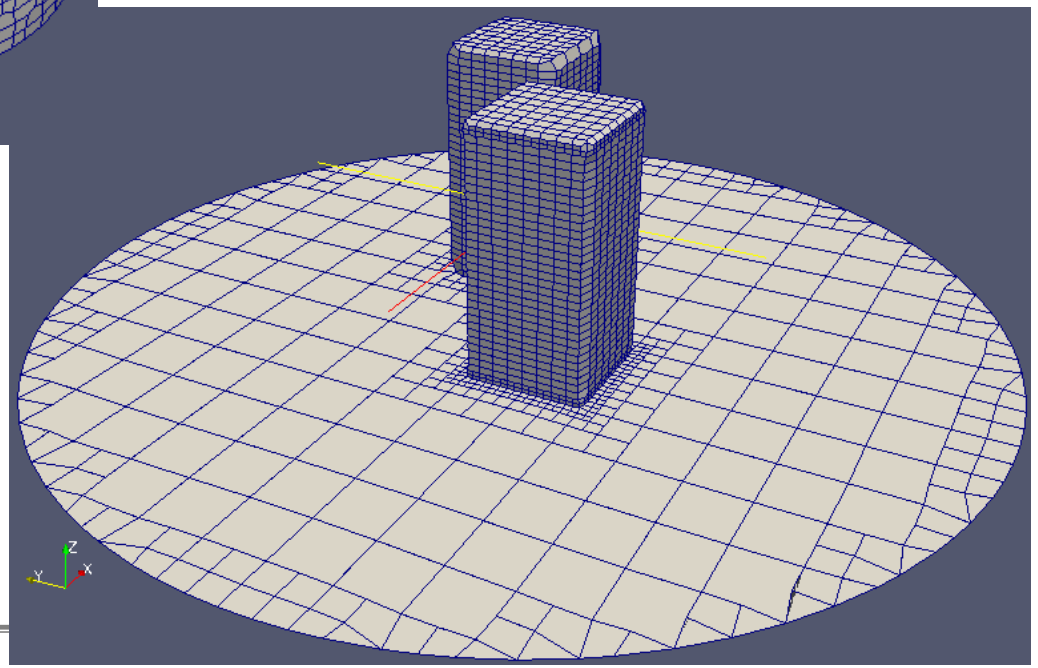
経緯

- iglooWithFridges例題をそのまま実行すると、冷蔵庫の角がまるまってしまう。
- その原因と対策を考える。
- それと同時に、基本形状の作成方法を学ぶ。




Z
Y
X


7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
-1.0
Y-Axis
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
-1.0


Z
Y
X

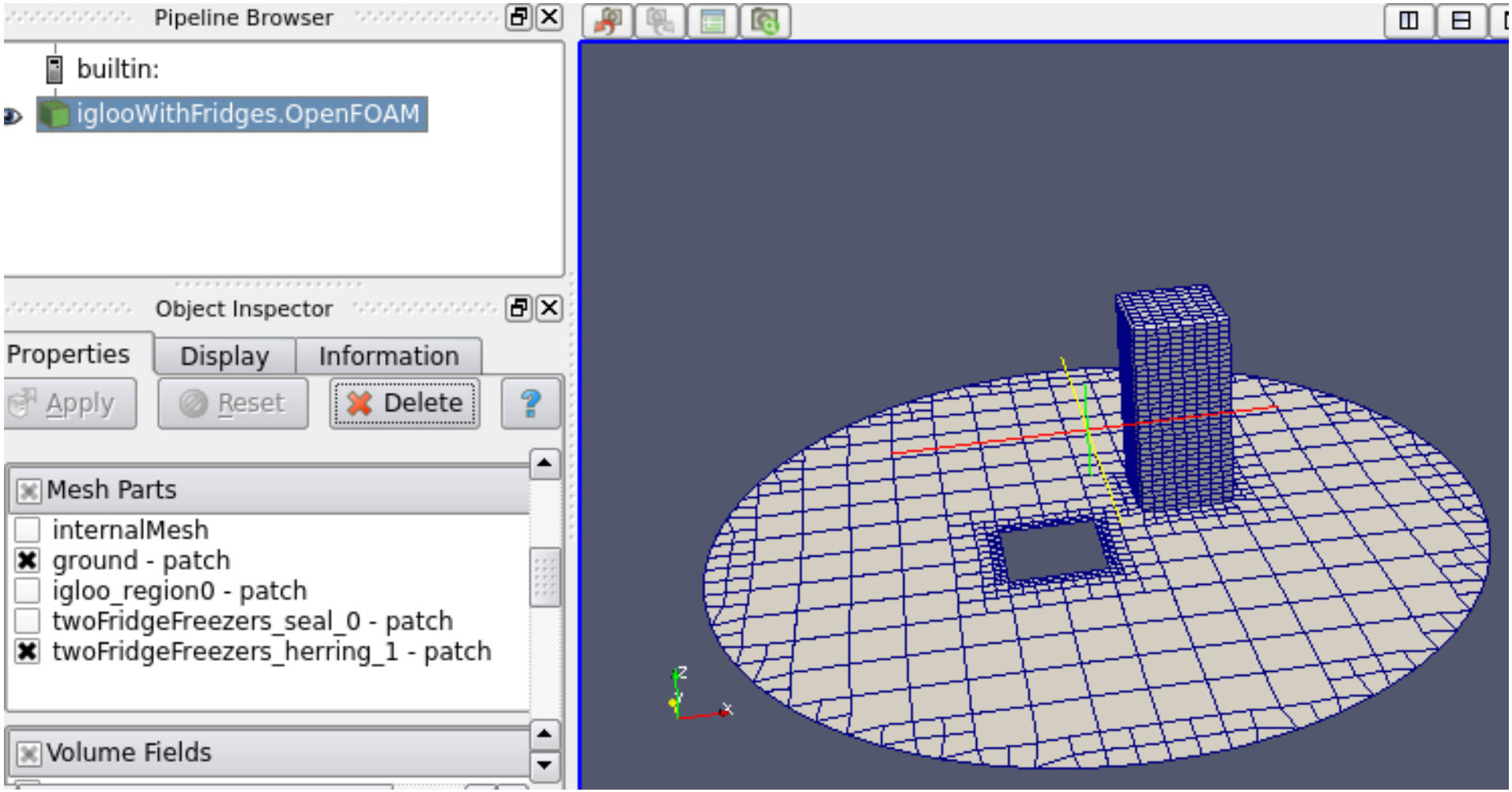
Original snappyHexMeshDict: geometry

```
geometry
{
    igloo
    {
        type searchableSphere;
        centre (3 3 0);
        radius 4;
    }

    box1
    {
        type searchableBox;
        min (0 0 0);
        max (1 1 1);
    }

    fridgeFreezer    //影響なし
    {
        type searchableSurfaceCollection;
        mergeSubRegions true;
        freezer
        {
            surface box1;
            scale (1 1 1);
            transform
            {
                type    cartesian;
                origin  (0 0 0);
                e1      (1 0 0);
                e3      (0 0 1);
            }
        }
        fridge
        {
            surface box1;
            scale (1 1 1.1);
            transform
            {
                type    cartesian;
                origin  (0 0 1);
                e1      (1 0 0);
                e3      (0 0 1);
            }
        }
    }

    twoFridgeFreezers    //これが使われる
    {
        type searchableSurfaceCollection;
        mergeSubRegions true;

        seal
        {
            surface box1;
            scale (1.0 1.0 2.1);
            transform
            {
                type    cartesian;
                origin  (2 2 0);
                e1      (1 0 0);    //local x axis
                e3      (0 0 1);    //local z axis
            }
        }
        herring
        {
            surface box1;
            scale (1.0 1.0 2.1);
            transform
            {
                type    cartesian;
                origin  (3.5 3 0);
                e1      (1 0 0);
                e3      (0 0 1);
            }
        }
    }
};
```

transform 設定について

http://foam.sourceforge.net/docs/cpp/a00321.html

- A coordinate rotation specified per local axes and the base class for other rotation specifications.
- The rotation is defined by a combination of local vectors (e1/e2), (e2/e3) or (e3/e1). Any nonorthogonality will be absorbed into the second vector.
- For convenience, the dictionary constructor forms allow a few shortcuts:
  - if the type is not otherwise specified, the type axes is implicit
  - if an axes specification (eg, e3/e1) is used, the **coordinateRotation** sub-dictionary can be dropped.
- Specifying the rotation by an **EulerCoordinateRotation** (type "EulerRotation") or by a **STARCDCoordinateRotation** (type "STARCDRotation") requires the **coordinateRotation** sub-dictionary.


Pipeline Browser
builtin:
iglooWithFridges.OpenFOAM
Object Inspector
Properties
Display
Information
Apply
Reset
Delete
Mesh Parts
internalMesh
ground - patch
igloo_region0 - patch
twoFridgeFreezers_seal_0 - patch
twoFridgeFreezers_herring_1 - patch
Volume Fields

castellatedMeshControls

```
// Settings for the castellatedMesh generation.
castellatedMeshControls
{

maxLocalCells 100000;

maxGlobalCells 2000000;

minRefinementCells 100;

nCellsBetweenLevels 1;

features
  (
    {
      file "fridgeA.eMesh";
      level 3;
    }
  );

  refinementSurfaces
  {
    twoFridgeFreezers
    {
      // Surface-wise min and max refinement level
      level (2 2);

      regions
      {
        // Region-wise override
        "cook.*"
        {
          level (3 3);
        }
      }
    }

    "iglo.*"
    {
      // Surface-wise min and max refinement level
      level (1 1);
    }
  }

resolveFeatureAngle 60;

  refinementRegions
  {
  }

locationInMesh (3 0.28 0.43);

allowFreeStandingZoneFaces true;
}
```

```
snapControls
{
nSmoothPatch 3;

tolerance 4.0;

nSolveIter 30;

nRelaxIter 5;
}



// Settings for the layer addition.
addLayersControls
{
relativeSizes true;

   // Per final patch (so not geometry!) the layer information
   layers
   {
      "two.*"
      {
         nSurfaceLayers 3;
      }
      "igloo_.*"
      {
         nSurfaceLayers 1;
      }
   }

expansionRatio 1.0;
finalLayerThickness 0.5;
minThickness 0.25;
nGrow 0;


featureAngle 60;
nRelaxIter 5;
nSmoothSurfaceNormals 1;
nSmoothNormals 3;
nSmoothThickness 10;
maxFaceThicknessRatio 0.5;
maxThicknessToMedialRatio 0.3;
minMedianAxisAngle 90;
nBufferCellsNoExtrude 0;
nLayerIter 50;
}
```

```
meshQualityControls
{
    //- Maximum non-orthogonality allowed. Set to 180 to disable.
    maxNonOrtho 65;

    //- Max skewness allowed. Set to <0 to disable.
    maxBoundarySkewness 20;
    maxInternalSkewness 4;

    //- Max concaveness allowed. Is angle (in degrees) below which concavity
    //  is allowed. 0 is straight face, <0 would be convex face.
    //  Set to 180 to disable.
    maxConcave 80;

    //- Minimum pyramid volume. Is absolute volume of cell pyramid.
    //  Set to a sensible fraction of the smallest cell volume expected.
    //  Set to very negative number (e.g. -1E30) to disable.
    minVol 1e-13;

    //- Minimum quality of the tet formed by the face-centre
    //  and variable base point minimum decomposition triangles and
    //  the cell centre.  Set to very negative number (e.g. -1E30) to
    //  disable.
    //     <0 = inside out tet,
    //      0 = flat tet
    //      1 = regular tet
    minTetQuality 1e-30;

    //- Minimum face area. Set to <0 to disable.
    minArea -1;

    //- Minimum face twist. Set to <-1 to disable. dot product of face normal
    //- and face centre triangles normal
    minTwist 0.05;

    //- minimum normalised cell determinant
    //- 1 = hex, <= 0 = folded or flattened illegal cell
    minDeterminant 0.001;

    //- minFaceWeight (0 -> 0.5)
    minFaceWeight 0.05;

    //- minVolRatio (0 -> 1)
    minVolRatio 0.01;

    //must be >0 for Fluent compatibility
    minTriangleTwist -1;


    // Advanced

    //- Number of error distribution iterations
    nSmoothScale 4;
    //- amount to scale back displacement at error points
    errorReduction 0.75;
}


// Advanced

// Flags for optional output
// 0 : only write final meshes
// 1 : write intermediate meshes
// 2 : write volScalarField with cellLevel for postprocessing
// 4 : write current intersections as .obj files
debug 0;


// Merge tolerance. Is fraction of overall bounding box of initial mesh.
// Note: the write tolerance needs to be higher than this.
mergeTolerance 1e-6;


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

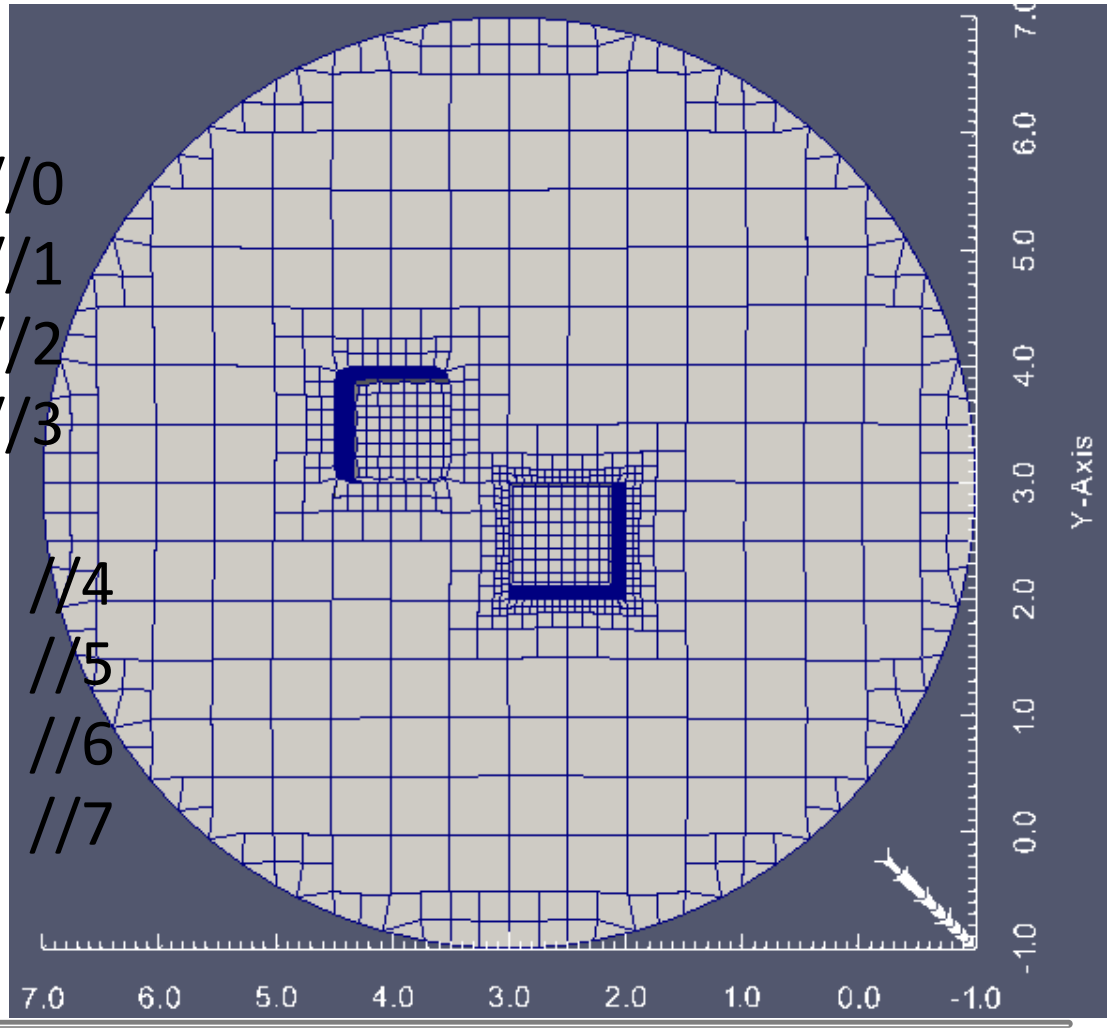
Problem 1

- constant/triSurface/fridgeA.eMesh ファイルに、特徴線を記述してある。
  - STLファイルを使うときは、surfaceFeatureExtractコマンドで作成した。
- eMesh の座標がおかしい？
  - geometryで指定している面のエッジとは、座標がことなる。
  - 修正する！

改良点

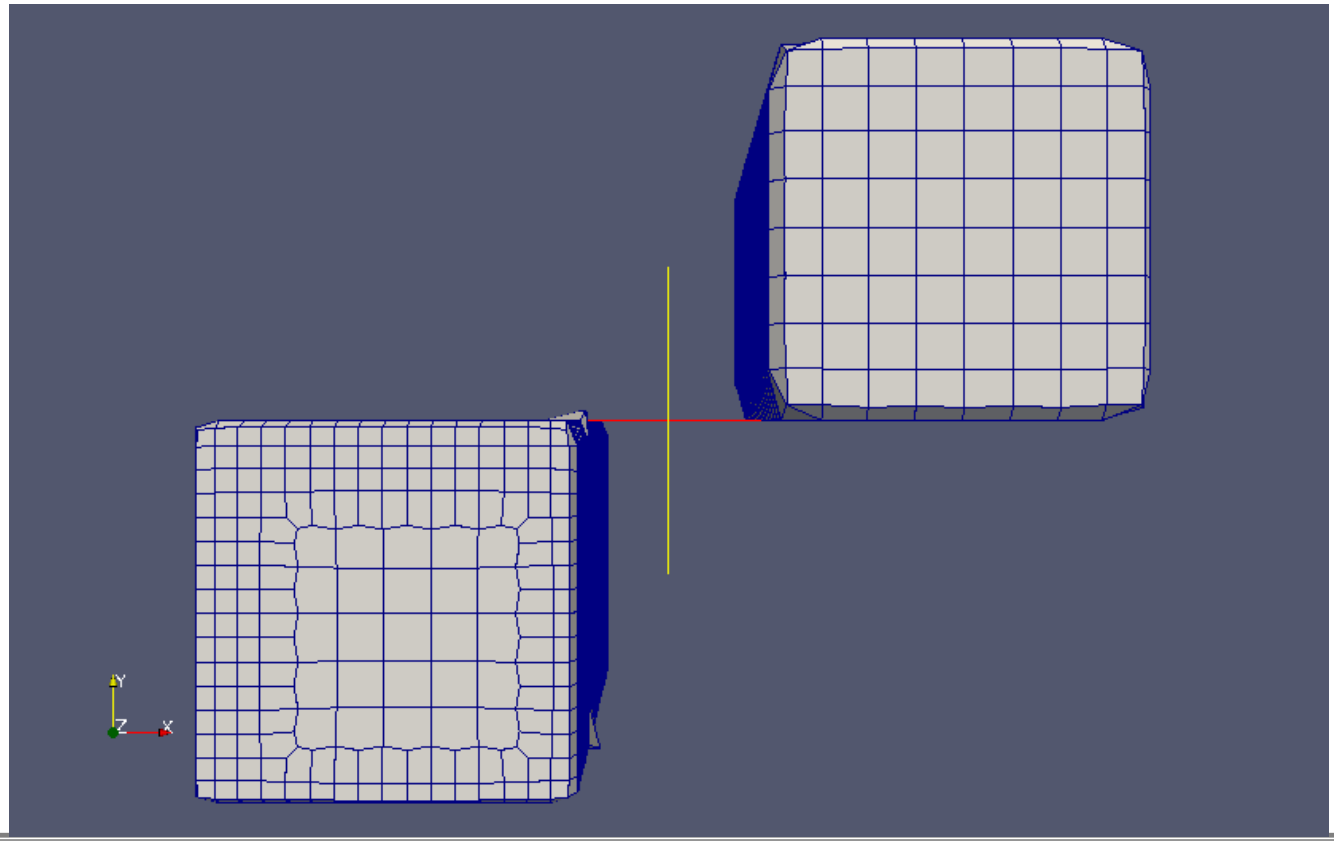
- fridgeA.eMesh の改善

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

(2 2 2.1) //(1.99 1.99 2.01)    //4
(3 2 2.1) //(3.01 1.99 2.01)    //5
(3 3 2.1) //(3.01 3.01 2.01)    //6
(2 3 2.1) //(1.99 3.01 2.01)    //7
)
```



効果　1

- エッジが少しシャープになる。Level3指定も効いている。（左：twoFridgeFreezers_seal_1）

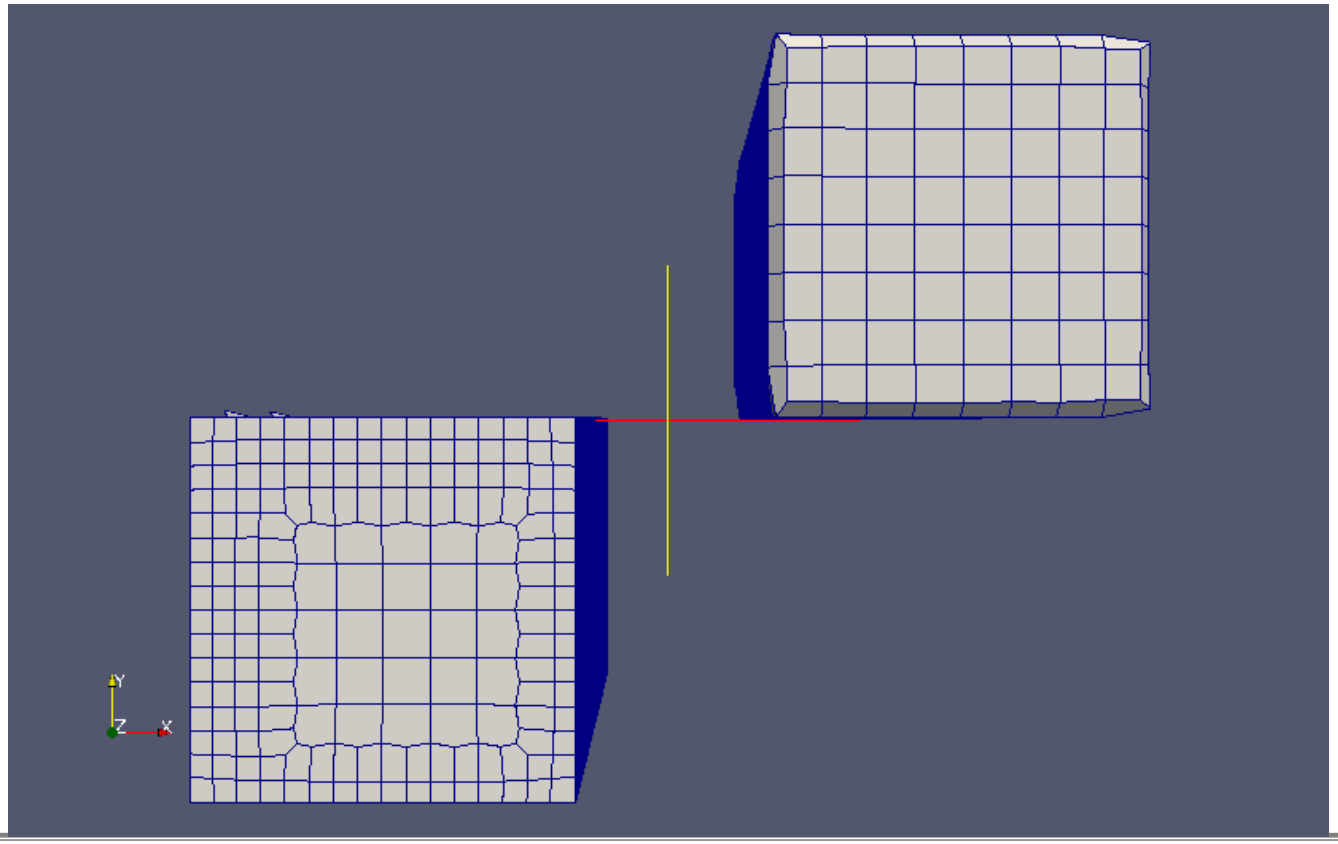


nFeatureSnapIter の指定

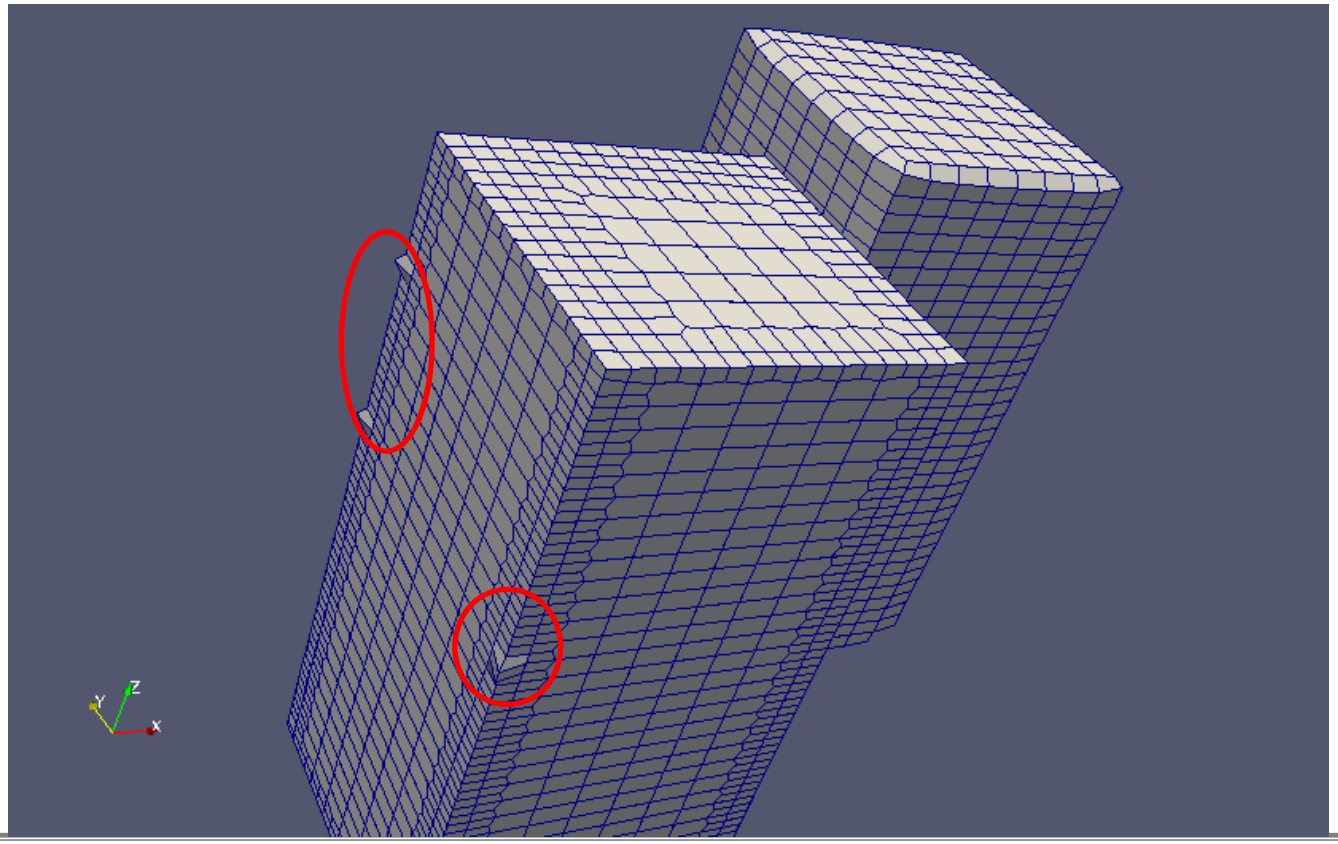
```
//- Highly experimental and wip: number of feature edge snapping
//  iterations. Leave out altogether to disable.
nFeatureSnapIter 10;
```

効果　2

- エッジがよりシャープになった。
- 不要な出っ張りはのこる。



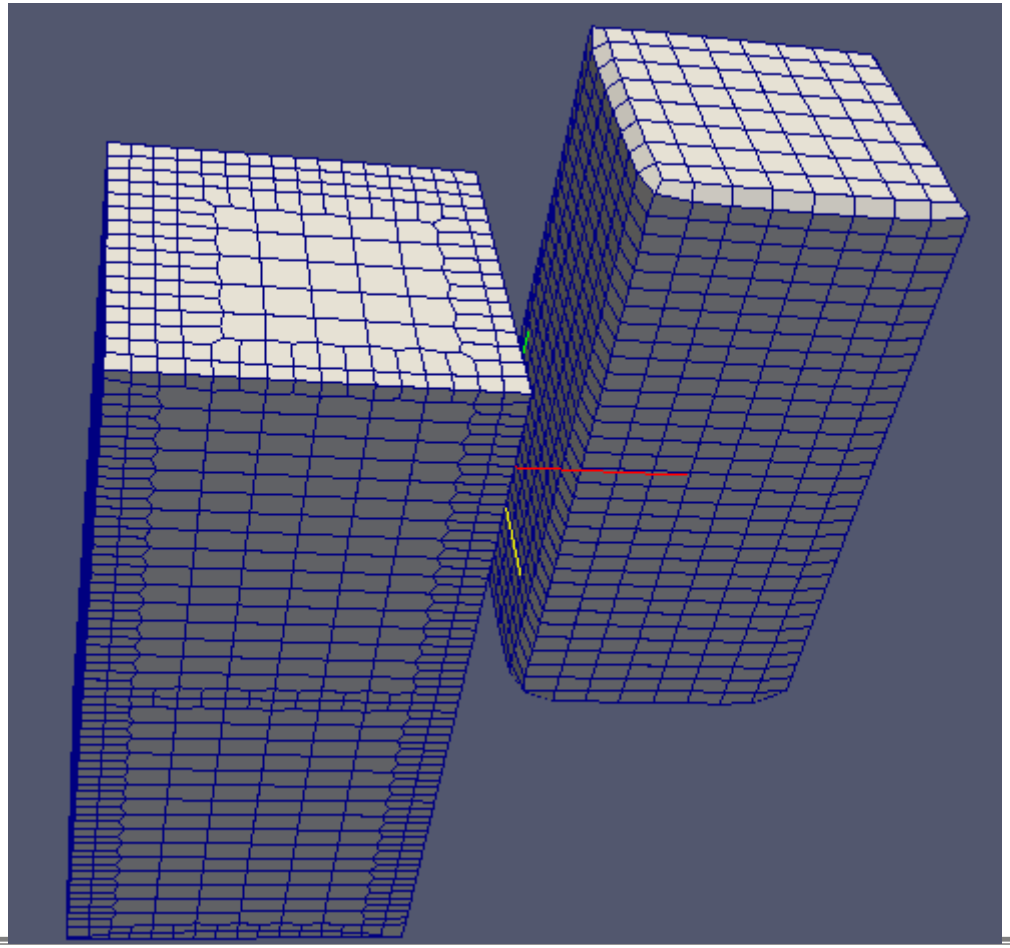
- 縦線部分に、でっぱりが。



layer の削除

効果３

- でっぱりがなくなる。



- Layerを入れて、かつ、エッジをきれいに再現するには？
- さらに調査を継続する。

snappyHexMeshを使うためのメモ

- STLファイルを使うときは、surfaceFeatureExtractを実行して、特徴線を抽出
- 形状をsnappyHexMeshDictに記述して生成するときは、それに応じた.eMeshファイルを自分で用意しておく。castellatedMeshControlsのfeaturesにファイル名を記述する。