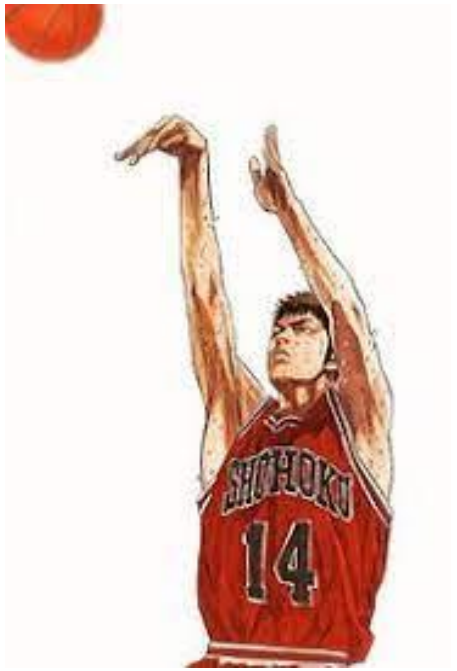
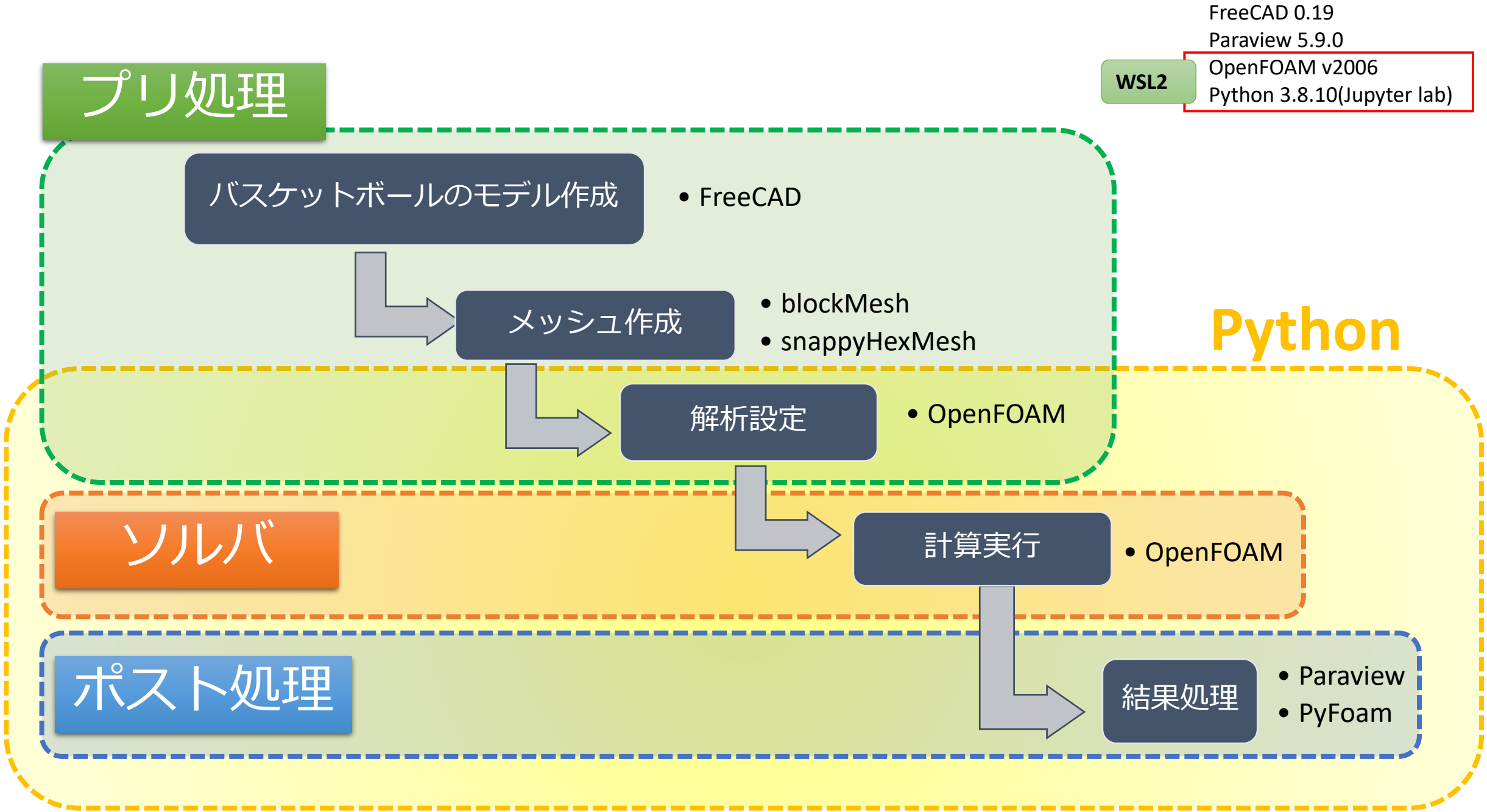
【OpenFOAM球体周りの抗力係数(3)】
snappyHexMeshで球体周りのメッシュ作成

2022年2月18日




FreeCAD 0.19
Paraview 5.9.0
WSL2
OpenFOAM v2006
Python 3.8.10(Jupyter lab)
プリ処理
バスケットボールのモデル作成
• FreeCAD
メッシュ作成
• blockMesh
• snappyHexMesh
解析設定
• OpenFOAM
Python
ソルバ
計算実行
• OpenFOAM
ポスト処理
結果処理
• Paraview
• PyFoam

今回のモデルは「20220216_sphere_coff_blog」というフォルダの中に作成します。

フォルダ構成

20220216_sphere_coff_blog

- model
- orgCase ←今回はこちらに球体周りのメッシュ作成
- resultDir

◘snappyHexMeshDictをコピー

(1) 「pitzDaily 」のチュートリアルにはsnappyHexMeshの設定ファイルがないため、 Terminal上で以下のように打ってls別のチュートリアルからファイルをコピーします。
「ls」コマンドでファイルがコピーされたかを確認できます。

```
$cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/motorBike/system/snappyHexMeshDict ./system/
```

```
[kamakiri@wsl]orgCase$cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/motorBike/system/snappyHexMeshDict ./system/
[kamakiri@wsl]orgCase$ls system/
blockMeshDict  controlDict  fvSchemes  fvSolution  snappyHexMeshDict  streamlines
[kamakiri@wsl]orgCase$
```

(2) 【OpenFOAM球体周りの抗力係数(1)】FreeCADで球体モデルを作るで作成した球体モデルを使うため、「constant」フォルダの中に「triSurface」というフォルダを作成し球体モデル（ball.stl)を保存します。

```
$mkdir constant/triSurface
$cp -r ../model/ball.stl constant/triSurface/
```

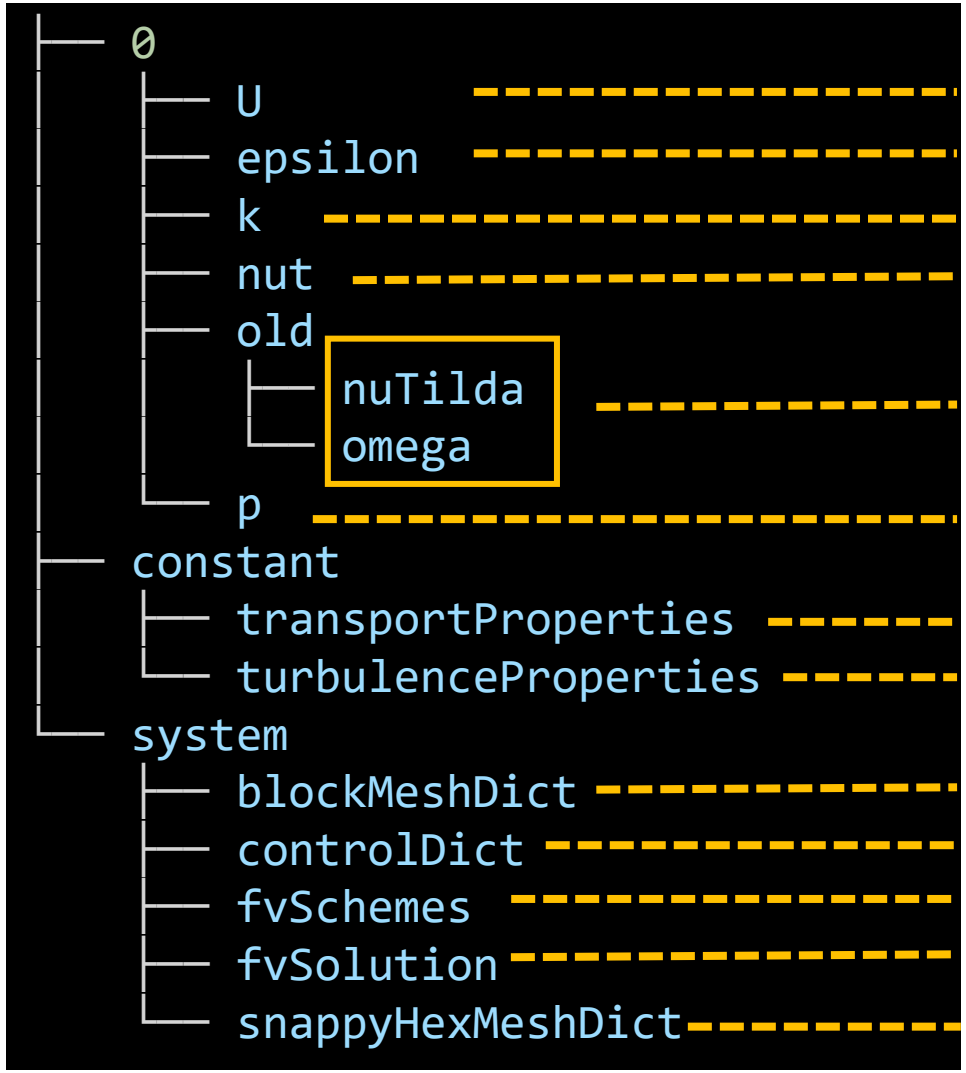
```
[kamakiri@wsl]orgCase$mkdir constant/triSurface
[kamakiri@wsl]orgCase$cp -r ../model/ball.stl constant/triSurface/
[kamakiri@wsl]orgCase$ls constant/triSurface/
ball.stl
[kamakiri@wsl]orgCase$
```

ロフォルダ構成

$tree

(0)Terminal上で
「tree」と打ってフォルダ構成を確認します。

※treeコマンドがインストールされていない場合はTerminalで「sudo apt install tree」と打ってインストールしてください。

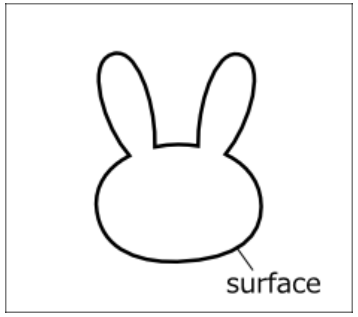


速度ベクトル
乱流消失率
乱流エネルギー
渦粘性係数
使わない
圧力場
物性値の設定
乱流モデルの指定
ベースメッシュの設定
実行制御
離散化スキームの設定
時間解法やマトリックスソルバの設定
メッシュ設定

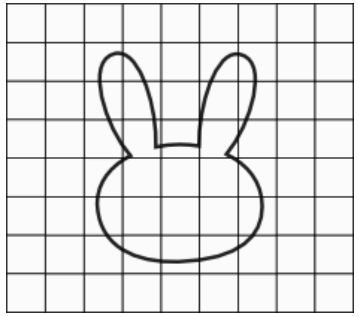
※blockMesh時に生成されたdynamicCodeなどは載せていません

◘snappyHexMeshの仕組み

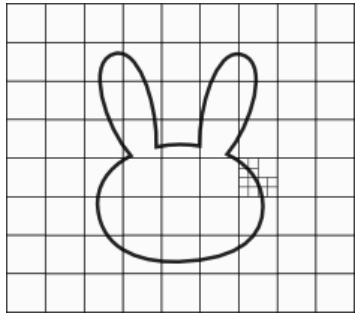
次のような手順でメッシュを作成する。



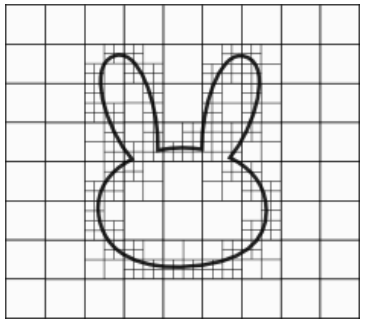
STLなどのサーフェスを用意。



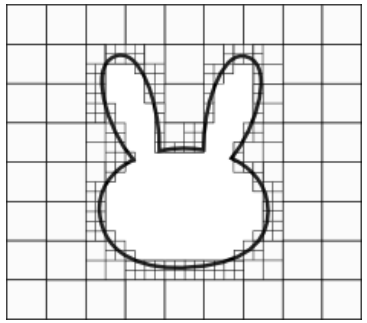
blockMeshでベースとなるメッシュを作成。



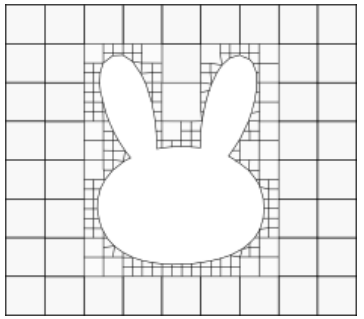
snappyHexMeshで境界近傍を六面体で細分化する。図では一部分だけ実施した。



サーフェイス全体に対して行う。



領域外になる部分のセルを取り除く。

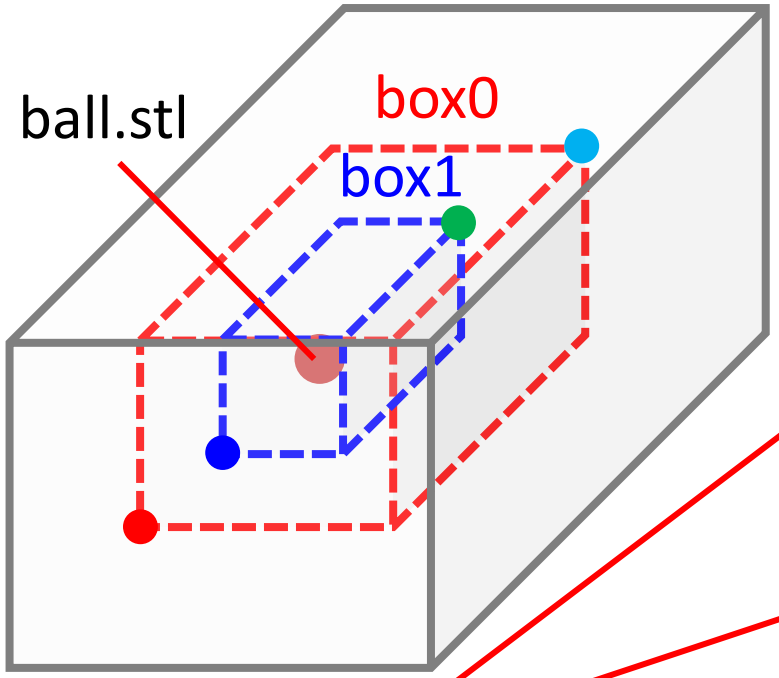


領域外に飛び出た部分をサーフェイスに合わせる ("snap")

■snappyHexMeshの設定

system/snappyHexMeshDict

を開いて編集を行う



ball.stl

orgCase > constant > triSurface > ball.stl

```
solid ball
  facet normal 0.109471 -0.069301 -0.991
    outer loop
      vertex 0.000000 -0.000000 -0.12500
      vertex 0.027268 -0.000000 -0.12199
      vertex 0.013445 -0.024746 -0.12178
    endloop
  endfacet
  facet normal 0.317167 -0.070870 -0.945
```

```
castellatedMesh true;
snap            true;
addLayers       true;境界層

geometry
{
  ball.stl   stlファイル
  {
    type triSurfaceMesh;
    name ball; stlファイル内の名前
               に対応させる
    regions
    {
      ball
      {
        name ball; 境界の名前
                   境界層で使う
      }
    }
  }
```

(3)境界の名前を設定

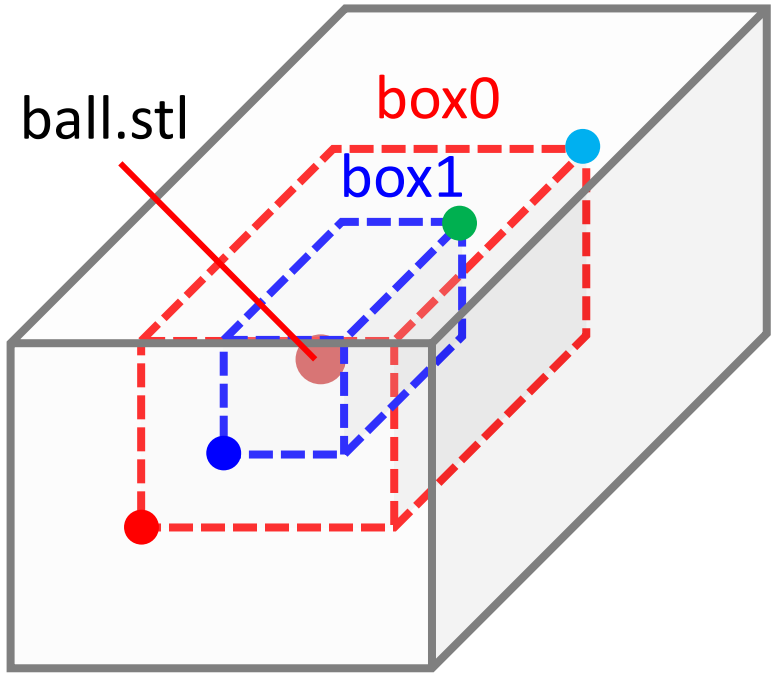
```
  box0
  {
    type searchableBox;
    min (-2.0 -0.6 -0.6);
    max (4.0 0.6 0.6);
  }
  box1
  {
    type searchableBox;
    min (-1.0 -0.41 -0.4);
    max (2.0 0.4 0.4);
  }
}
```

(4)メッシュ細分化領域の指定

◘snappyHexMeshの設定

system/snappyHexMeshDict

を開いて編集を行う



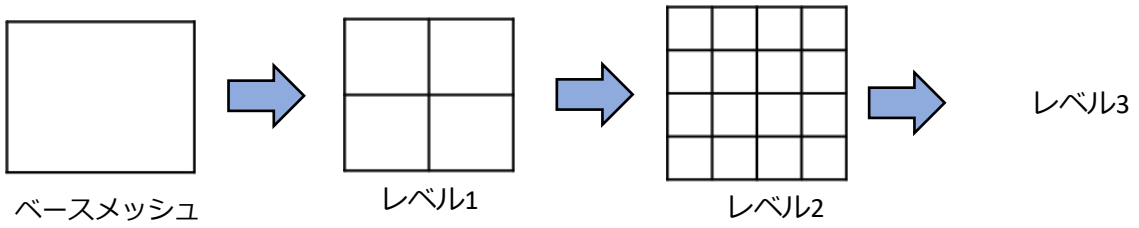

```
castellatedMeshControls
{
    maxLocalCells 100000;
    maxGlobalCells 2000000;
    minRefinementCells 10;
    maxLoadUnbalance 0.10;
    nCellsBetweenLevels 3;
    features();
    refinementSurfaces
    {
        ball
        {
            level (0 0);
            patchInfo
            {
                type wall;
            }
        }
    }
```

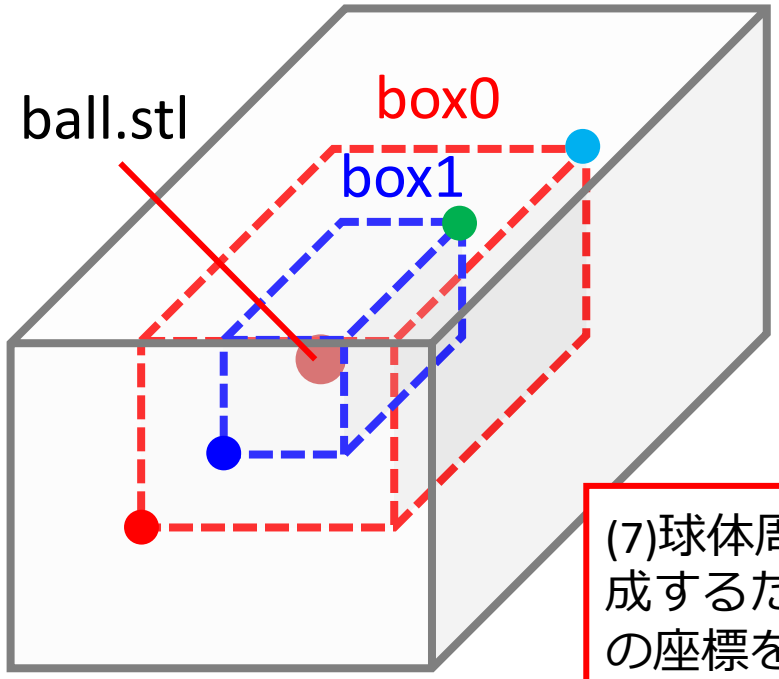
(5)球体の境界面タイプを指定今回は壁にしたいのでwallとする

◘snappyHexMeshの設定

system/snappyHexMeshDict

を開いて編集を行う





(7)球体周りにメッシュ作成するため、球体より外側の座標を指定

```
    // Resolve sharp angles
    resolveFeatureAngle 30;
    refinementRegions
    {
        box0
        {
            mode inside;
            levels ((3 2));
        }
        box1
        {
            mode inside;
            levels ((3 3));
        }

    locationInMesh (2.0 0.0 0.0);
    allowFreeStandingZoneFaces true;
}
```

(6)メッシュの再分割レベルの指定

再分割レベル

再分割レベル

メッシュ生成部分の指定

◘snappyHexMeshの設定

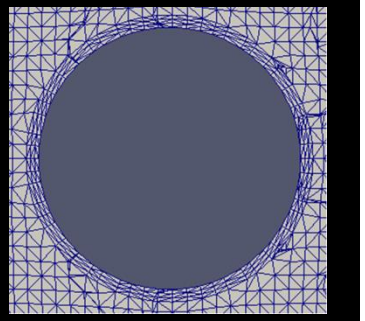
system/snappyHexMeshDict

を開いて編集を行う

```
// Settings for the snapping.
snapControls
{
    nSmoothPatch 3;
    tolerance 2.0;
    nSolveIter 30;
    nRelaxIter 5;
    nFeatureSnapIter 10;
    implicitFeatureSnap false;
    explicitFeatureSnap true;
    multiRegionFeatureSnap false;
}
```

```
addLayersControls
{
    relativeSizes true;
    layers
    {
        "ball"
        {
        nSurfaceLayers 5;
        }
    }
    expansionRatio 1.0;
    finalLayerThickness 0.3;
    minThickness 0.1;
    nGrow 0;
    featureAngle 60;
    slipFeatureAngle 30;
    nRelaxIter 3;
    nSmoothSurfaceNormals 1;
    nSmoothNormals 3;
    nSmoothThickness 10;
    maxFaceThicknessRatio 0.5;
    maxThicknessToMedialRatio 0.3;
    minMedianAxisAngle 90;
    nBufferCellsNoExtrude 0;
    nLayerIter 50;
}
```

(8)境界層メッシュを作成する面の指定。
今回は5層にする。



レイヤーメッシュ
物体表面では流れが遅くなり、表面から離れるにしたがって流れが速くなる。また、表面に沿って流れが分布されるため流体の速度変化をとらえるためのメッシュの層を挿入することが推奨される。

◘snappyHexMeshの設定

system/snappyHexMeshDict

を開いて編集を行う

```
meshQualityControls
{
    #include "meshQualityDict"
}
```

(9)「system」の中の「meshQualityDict」ファイル読み込む。
meshQualityDictは別のチュートリアルからコピーしてくる。
Terminal上で以下のように打って、systemにコピーする。

```
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/motorBike/system/meshQualityDict    ./system/
```

system/meshQualityDict

```
// Include defaults parameters from master dictionary
#includeEtc "caseDicts/mesh/generation/meshQualityDict"

//- minFaceWeight (0 -> 0.5)
minFaceWeight 0.02;
```

さらにincludeしているので下記のようにしてファイルの中身を確認できる。
ここでは割愛。

```
vi  $FOAM_ETC/caseDicts/mesh/generation/meshQualityDict
```

◘並列計算の設定

(9)「system」の中の「decomposeParDict 」がないため、別のチュートリアルからコピーしてくる。Terminal上で以下のように打って、systemにコピーする。

```
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/motorBike/system/decomposeParDict    ./system/
```

system/decomposeParDict

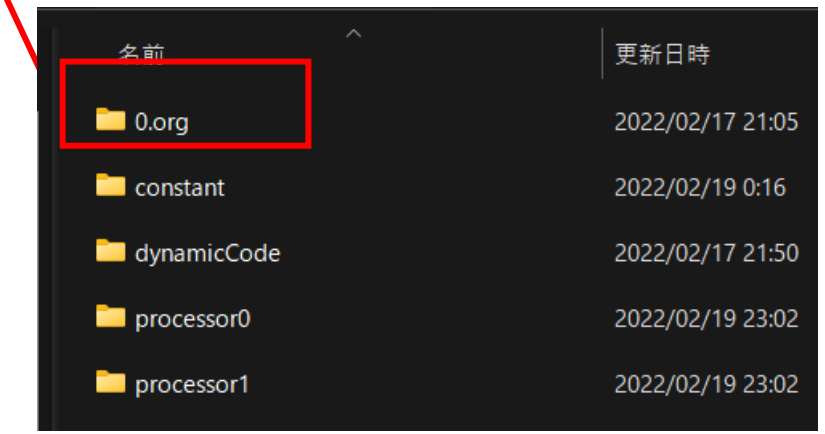
を開いて編集を行う

(10)並列数を4にし、methodを「scotch」に変更。scotchにすると適当に分割してくれるので

methodをscotchにしたのでこちらは使っていない

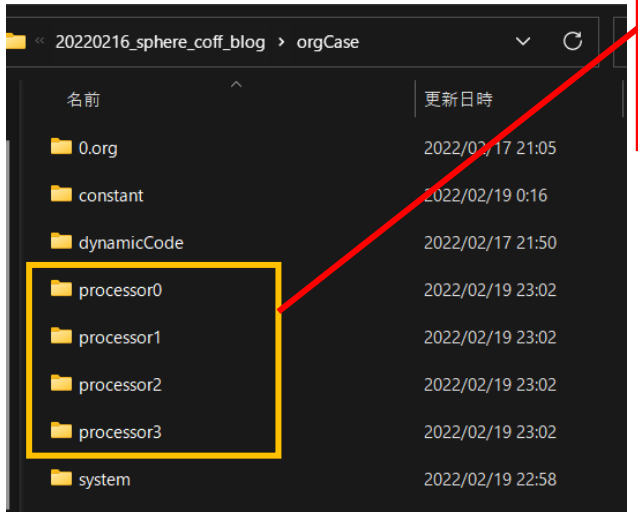
```
numberOfSubdomains 4;

method          scotch;
//method          hierarchical;
// method          ptscotch;

simpleCoeffs
{
    n                   (4 1 1);
    delta               0.001;
}

hierarchicalCoeffs
```

(11)ファイル名が「０」だとsnappyHexMeshでエラーが出るため「0.org」に変更しておく。



◘snappyHexMeshの実行

```
decomposePar
```



(12)計算業域の分割を行う
並列数が4なのでフォルダが4つに分解されている
ことを確認

(13)snappyHexMeshの実行

```
$mpirun -np 4 snappyHexMesh -parallel
```

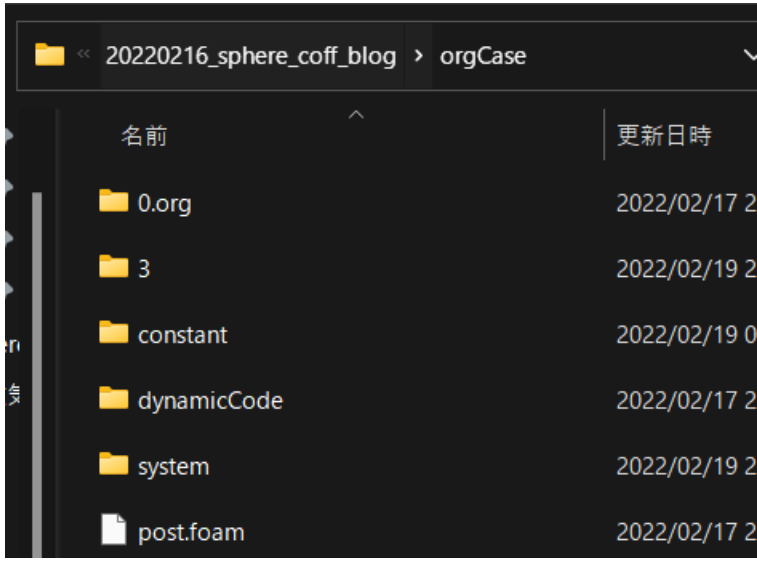
(14)分割した計算領域を結合

```
$reconstructParMesh -latestTime -mergeTol 1e-6
```

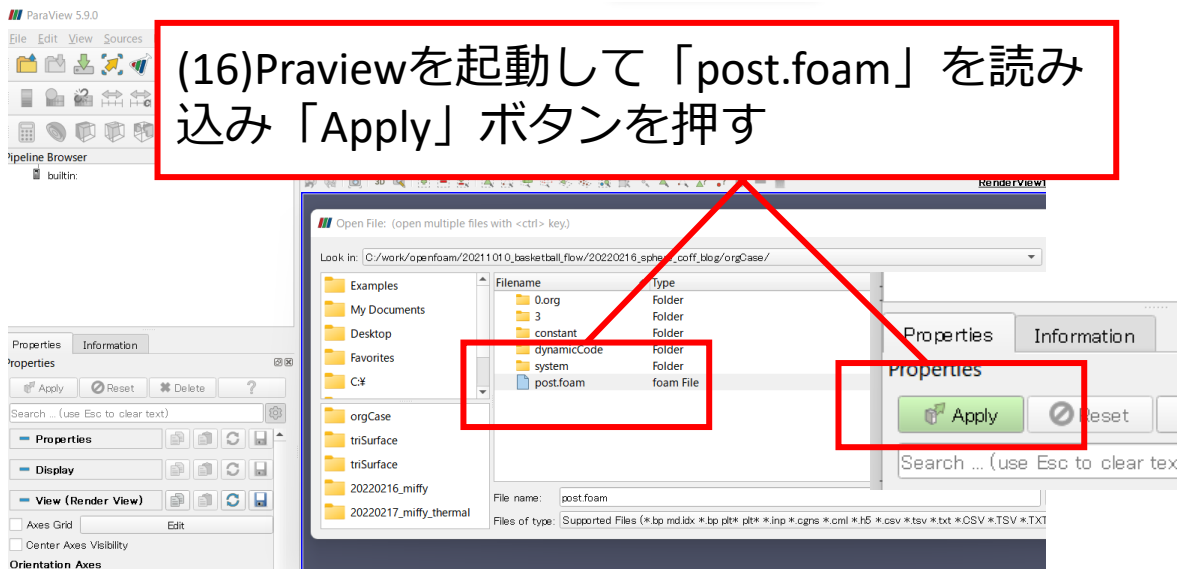
(15)分割したフォルダは使わないので削除

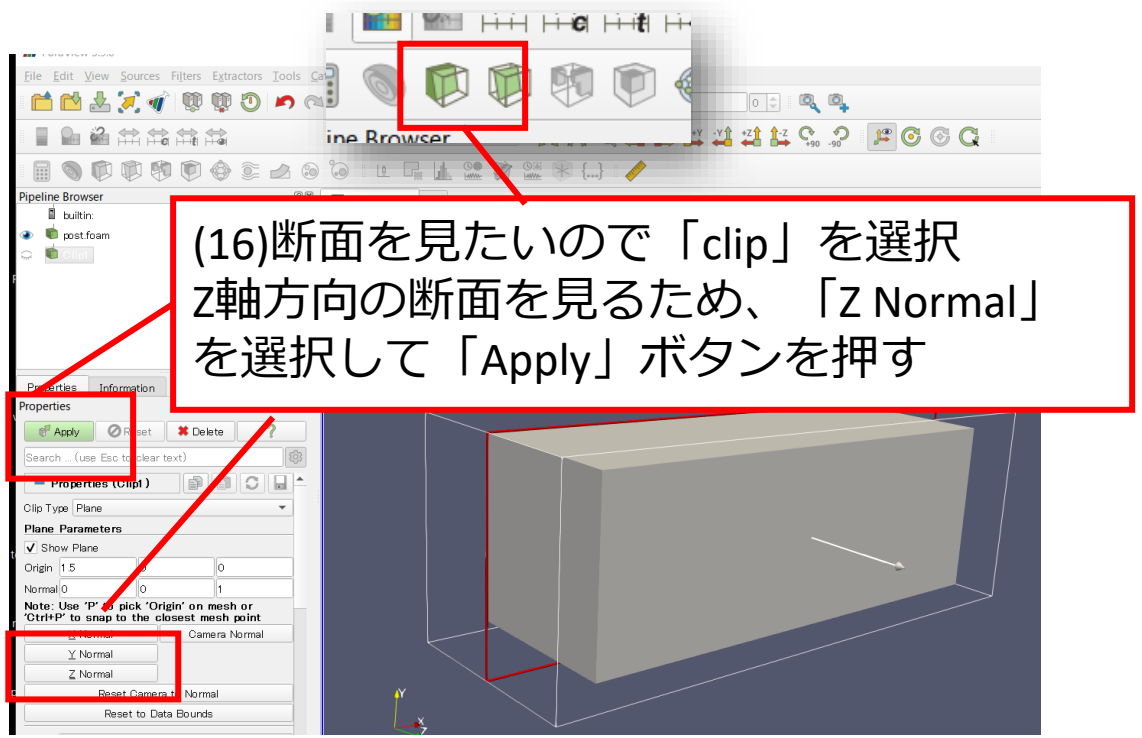
```
$rm -r processor*
```

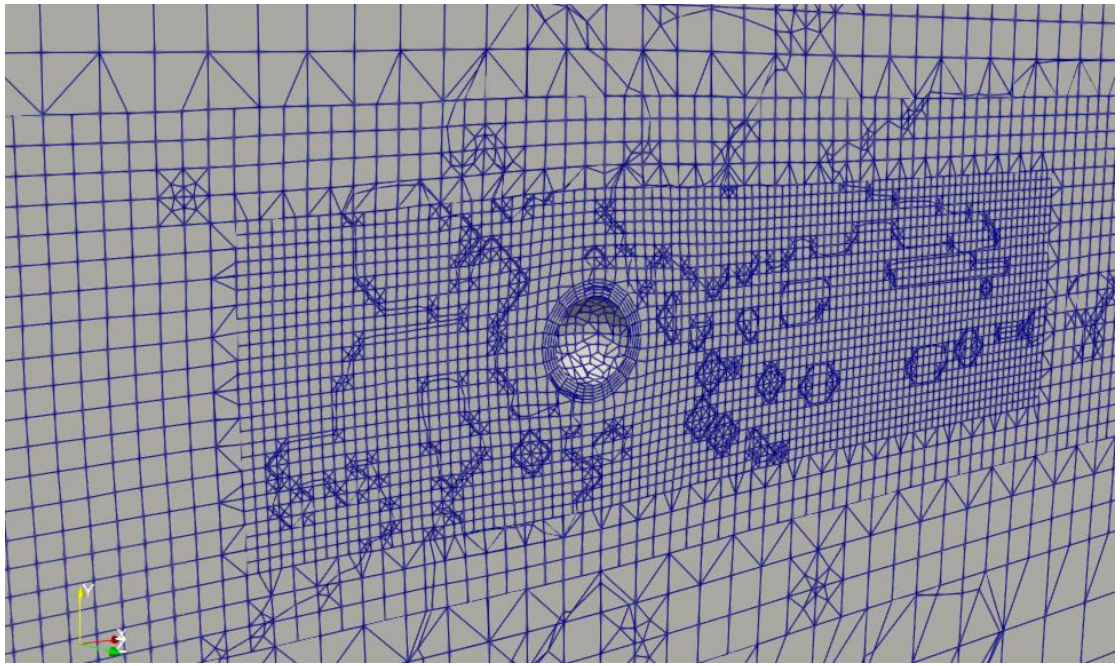
(15)までの操作後のフォルダ内



◘Paraviewで結果を確認







断面がこのようになっていればOK
境界層メッシュも5層あるか確認

おわり