

# 説明内容

1. gridEditorのField内変数の扱い方
2. multiRegionモデルにレイヤを追加する方法

# 1. gridEditorでField内変数を取り扱う

Fieldで扱う変数は、

include、inGroupsでも定義しており複雑。 一見して、内容が解らない。

tutorialsの例:tutorials/compressible/rhoPimpleDyMFoam/annularThermalMixer

```
boundary
16 // *****
17
18 12
19 (
20   innerInlet
21   {
22       type            patch;
23       inGroups         1(inlet);
24       nFaces           544;
25       startFace        245836;
26   }
27   outerInlet
28   {
29       type            patch;
30       inGroups         1(inlet);
31       nFaces           1404;
32       startFace        246380;
33   }
34   innerOutlet
35   {
36       type            patch;
37       inGroups         1(outlet);
38       nFaces           54;
39       startFace        24;
40   }
41   outerOutlet
```

```
13 T field
14 }
15 // *****
16
17 #include "${FOAM_CASE}/constant/caseSettings"
18
19 dimensions      [0 0 0 1 0 0 0];
20
21 internalField    uniform $:outerInlet.T;
22
23 boundaryField
24 {
25     innerInlet
26     {
27         type            fixedValue;
28         value            uniform $:innerInlet.T;
29     }
30     outerInlet
31     {
32         type            fixedValue;
33         value            uniform $:outerInlet.T;
34     }
35     outlet            { $:outlet.T; }
36     staticWalls { $:wall.T; }
37     movingWalls { $staticWalls; }
38     #include "${WM_PROJECT_DIR}/etc/caseDicts/setConstraintTypes"
39 }
40 }
41
42 // *****
```

includeファイル

変数 \$:outerInlet.T

inGroups

## includeファイル: caseSettings

```

12     class      dictionary,
13     location    "";
14     object      caseSummary;
15 }
16 // *****
17
18 innerInlet
19 {
20     U            (0 0 0.2);
21     epsilon      5.70e-5;
22     k            2.40e-5;
23     T            233;
24 }
25
26 outerInlet
27 {
28     U            (0 0 0.1);
29     epsilon      3.98e-5;
30     k            6.00e-6;
31     T            293;
32 }
33
34 meshMotionProperties
35 {
36     omega 25; // rad/s
37 }
38
39 #include "${FOAM_CASE}/constant/boundaryConditions"
40
41 // ***** includeファイル *****

```

変数 `$.outerInlet.T` の内容

## includeファイル: boundaryConditions

```

boundaryConditions
--
39
40 outlet
41 {
42     p
43     {
44         type      totalPressure;
45         value      uniform 1e5;
46         p0         uniform 1e5;
47         U          U;
48         phi        phi;
49         rho        rho;
50         psi        none;
51         gamma      1.4;
52     }
53
54     U
55     {
56         type      pressureInletOutletVelocity;
57         value      uniform (0 0 0);
58     }
59
60     T
61     {
62         type      inletOutlet;
63         inletValue uniform $.outerInlet.T;
64         value      $inletValue;
65     }
66
67     k
68     {

```

patch内容 `$.outerInlet.T` の内容

# gridEditorでcaseを開いた状態










gridEditor: annularThermalMixer/0/. (0:1)

ファイル(F) 編集(E) 表示(V)

field内変数が全て解析できておらず、空白が発生

	define patch at constant/. (boundary)	T	U	alphan	
field type dimensions		volScalarField; [0 0 0 1 0 0 0];	volVectorField; [0 1 -1 0 0 0 0];	volScalarField; [1 -1 -1 0 0 0 0];	volScalarField; [0 2 -3 0 0 0 0];
internal Field		uniform \$:outerInlet.T;	uniform (0 0 0);	uniform 0;	uniform \$:innerInlet.T;
innerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform \$:innerInlet.T;	type fixedValue; value uniform \$:innerInlet.U;	\$:calculated.alphan;	type fixedValue; value uniform \$:innerInlet.T;
outerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform \$:outerInlet.T;	type fixedValue; value uniform \$:outerInlet.U;	\$:calculated.alphan;	type fixedValue; value uniform \$:outerInlet.T;
innerOutlet	type patch; inGroups 1(outlet);	patch内容が変数のまま		\$:calculated.alphan;	\$:outlet.epsilon;
outerOutlet	type patch; inGroups 1(outlet);	\$:outlet.T;	\$:outlet.U;	\$:calculated.alphan;	\$:outlet.epsilon;
rotorBlades	type wall; inGroups 2 ( movingWalls wall );	fieldのpatchの定義が無いので、inGroupsを読みに行くが inGroupsの「movingWalls」が読めず、空白のまま			
	type wall; inGroups	inGroups 2(movingWalls wall) movingWalls{\$staticWalls} staticWalls {\$:wall.T}			

## gridEditor側で変数の参照を判断し表示できるように修正

gridEditor: annularThermalMixer/0/. (0:1)				
ファイル(F) 編集(E) 表示(V)				
        				
	define patch at constant/. (boundary)	T	U	alphan
field type dimensions		volScalarField; [0 0 0 1 0 0 0];	volVectorField; [0 1 -1 0 0 0 0];	volScalarField; [1 -1 -1 0 0 0 0];
internal Field		uniform 293;	uniform (0 0 0);	uniform 0;
innerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform 233;	type fixedValue; value uniform (0 0 0.2);	type calculated; value uniform 0;
outerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform 293;	type fixedValue; value uniform (0 0 0.1);	type calculated; value uniform 0;
innerOutlet	type patch; inGroups 1(outlet);	type inletOutlet; inletValue uniform 293; value <u>\$inletValue</u> ;	type pressureInletOutletVelocity; value uniform (0 0 0);	type calculated; value uniform 0;
outerOutlet	type patch; inGroups 1(outlet);	type inletOutlet; inletValue uniform 293; value \$inletValue;	type pressureInletOutletVelocity; value uniform (0 0 0);	type calculated; value uniform 0;

この形式の変数は、そのまま残す

## gridEditorでfieldを保存した結果

```
15 // ****
16
17 #include "${FOAM_CASE}/constant/caseSettings"
18
19 dimensions      [0 0 0 1 0 0 0];
20
21 internalField    uniform 293;
22
23 boundaryField
24 {
25     #include "${WM_PROJECT_DIR}/etc/caseDicts/setConstraintTypes"
26
27     innerInlet
28     {
29         type      fixedValue;
30         value      uniform 293;
31     }
32     outerInlet
33     {
34         type      fixedValue;
35         value      uniform 293;
36     }
37     innerOutlet
38     {
39         type      inletOutlet;
40         inletValue uniform 293;
41         value      $inletValue;
42     }
43     outerOutlet
44     {
45         type      inletOutlet;
```

includeファイルは残る

boundaryField内のincludeファイルは、  
boundaryField内の最上位に移動

patch内容を判断した結果を保存する。



# 通常の変数をgridEditorで扱う

	define patch at constant/. (boundary)	T	
field type dimensions		volScalarField; [0 0 0 1 0 0 0];	vo [0
internal Field		uniform 293;	un
innerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform 233;	ty va
outerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform 293;	ty va
innerOutlet t	type patch; inGroups 1(outlet);	type inletOutlet; <u>inletValue uniform 293;</u> value \$inletValue;	ty va
outerOutlet t	type patch; inGroups 1(outlet);	type inletOutlet; <u>inletValue uniform 293;</u> value \$inletValue;	ty va
rotorBlade	type wall; inGroups 2	type zeroGradient;	ty va

同じ変数を2箇所を設定

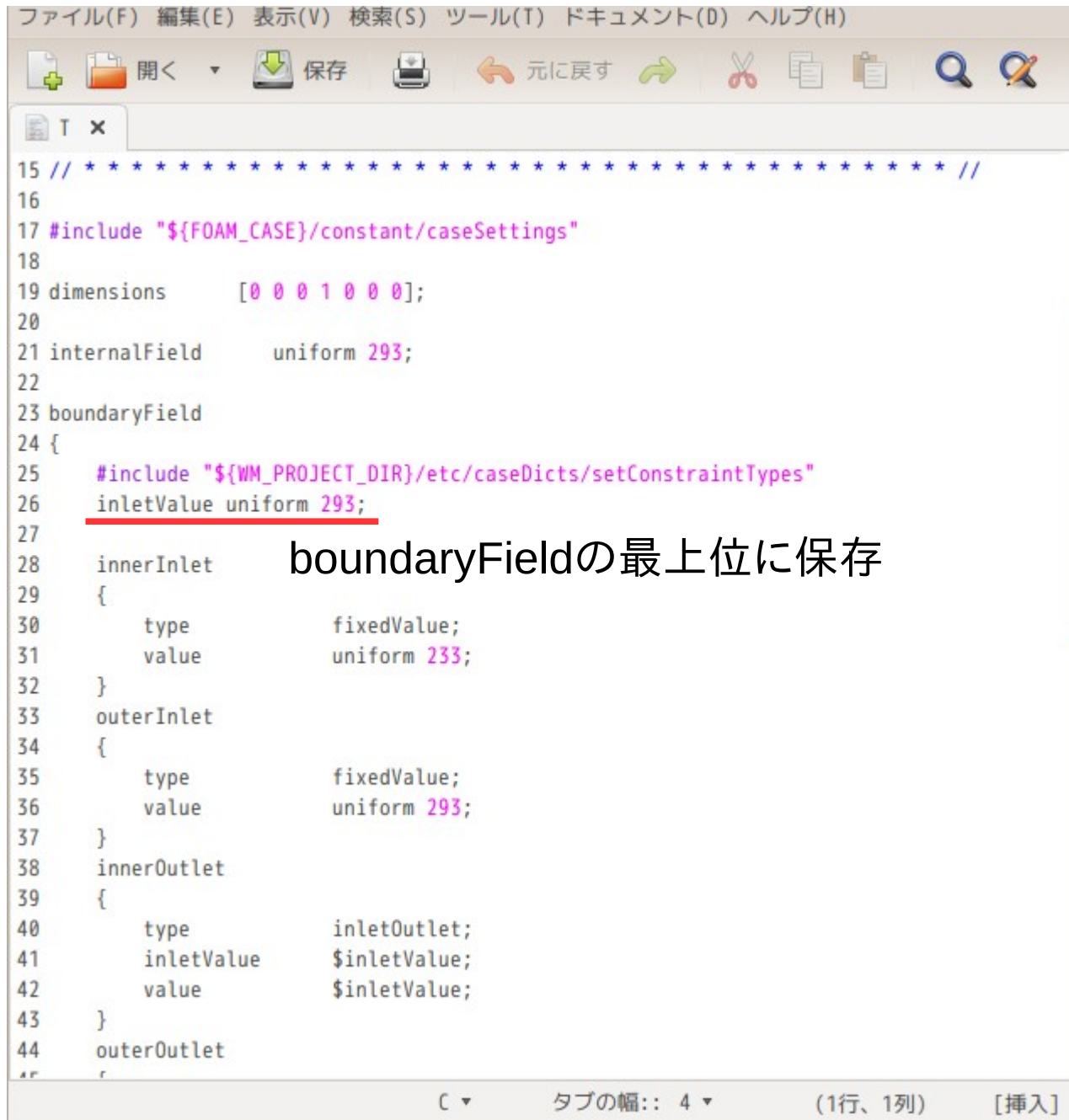
	define patch at constant/. (boundary)	T	
field type dimensions		volScalarField; [0 0 0 1 0 0 0];	vo [0
		uniform 293;	un
otherNames (boundary)		inletValue uniform 293;	
innerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform 233;	ty va
outerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform 293;	ty va
innerOutlet t	type patch; inGroups 1(outlet);	type inletOutlet; inletValue \$inletValue; value \$inletValue;	ty va
outerOutlet	type patch; inGroups 1(outlet);	type inletOutlet; inletValue \$inletValue; value \$inletValue;	ty va

変数行を追加

変数「inletValue uniform 293」を外に出す  
一箇所の変更で、該当箇所の値が変更できる

	type wall;	type zeroGradient;	ty
--	------------	--------------------	----

## 変数行を保存した結果



```
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
開く 保存 元に戻す
T x
15 // ****
16
17 #include "${FOAM_CASE}/constant/caseSettings"
18
19 dimensions      [0 0 0 1 0 0 0];
20
21 internalField    uniform 293;
22
23 boundaryField
24 {
25     #include "${WM_PROJECT_DIR}/etc/caseDicts/setConstraintTypes"
26     inletValue uniform 293;
27
28     innerInlet    boundaryFieldの最上位に保存
29     {
30         type      fixedValue;
31         value      uniform 233;
32     }
33     outerInlet
34     {
35         type      fixedValue;
36         value      uniform 293;
37     }
38     innerOutlet
39     {
40         type      inletOutlet;
41         inletValue $inletValue;
42         value      $inletValue;
43     }
44     outerOutlet
45     {
```

C ▾ タブの幅:: 4 ▾ (1行、1列) [挿入]



# 変数を入力する方法

gridEditor: annularThermalMixer\_copy0/0/. (0:0)

ファイル(F) 編集(E) 表示(V)

	define patch at constant/. (boundary)	T	
field type dimensions		volScalarField; [0 0 0 1 0 0 0];	volVectorF [0 1 -1 0 0 0];
internal Field		uniform 293;	uniform (0

行ラベルを右クリックして  
「変数定義行(空)の表示/非表示切り替え」  
を選択して定義行を表示させる

行コピー  
行貼付  
patch名sortする/しない切替え  
cell内の表示行数・データ数変更  
patch名変更  
新しい空patch追加  
空patch削除  
全ての空patch削除  
変数定義行(空)の表示/非表示切り替え

gridEditor: annularThermalMixer\_copy0/0/. (0:2)

ファイル(F) 編集(E) 表示(V)

	define patch at constant/. (boundary)	T	
field type dimensions		volScalarField; [0 0 0 1 0 0 0];	vol [0
otherNames			
internal Field		uniform 293;	uni
otherNames (boundary)		inletValue uniform 293;	
innerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform 293;	typ val
outerInlet	type patch; inGroups 1(inlet);	type fixedValue; value uniform 293;	typ val
innerOutle t	type patch; inGroups 1(outlet);	type inletOutlet; inletValue \$inletValue; value \$inletValue;	typ val
outerOutle	type patch; inGroups 1(outlet);	type inletOutlet; inletValue \$inletValue; value \$inletValue;	typ val

変数定義行が表示されるので、入力できる

変数を解釈する方法は、今回一例で確認したばかり  
他のcaseでも対応できるか、今後確認

## multiRegionモデルの流体側にレイヤを作成する

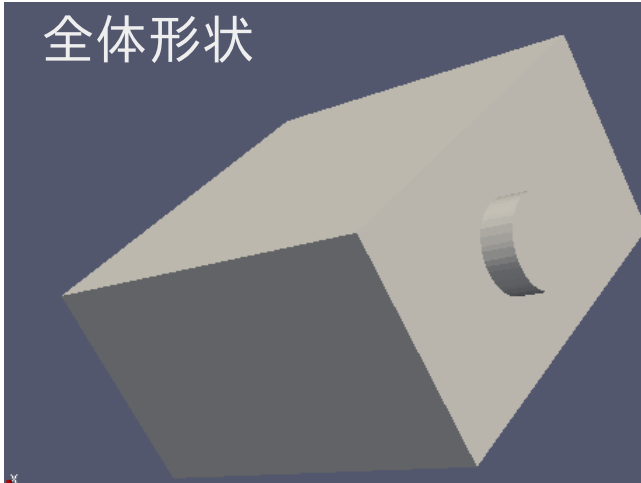
snappyHexMeshでは、  
patchには、レイヤを付ける事ができるが、  
流体・個体境界面には付けられない。



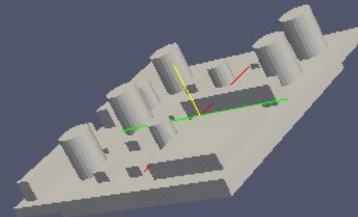
先にメッシュを作り、領域分割した後に、流体部のメッシュを抜き出し、  
snappyHexMeshでレイヤのみ追加する事で  
流体部分に、レイヤを追加できる。

# multiRegionモデル

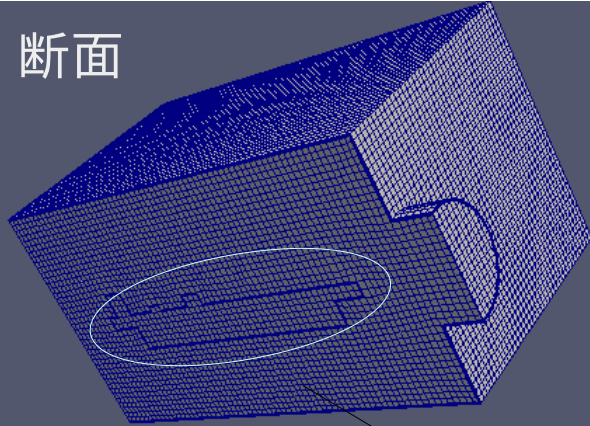
全体形状



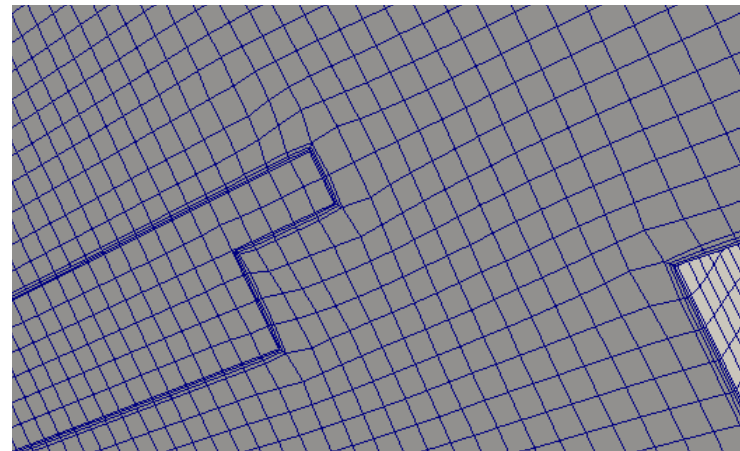
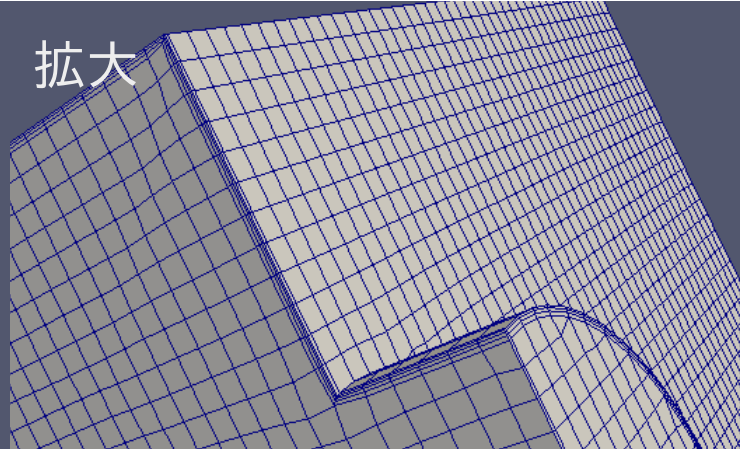
内部部品 (個体部)



断面



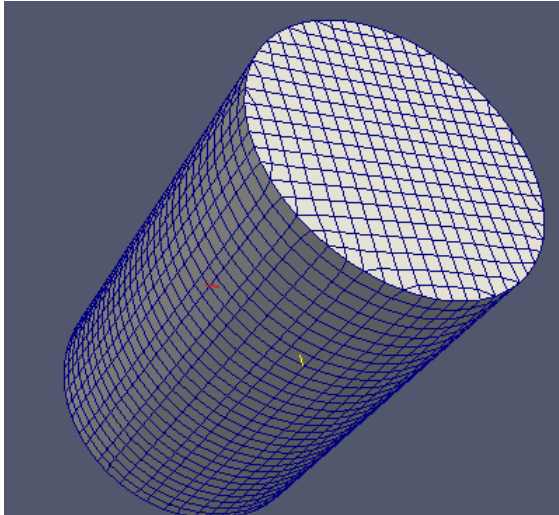
拡大



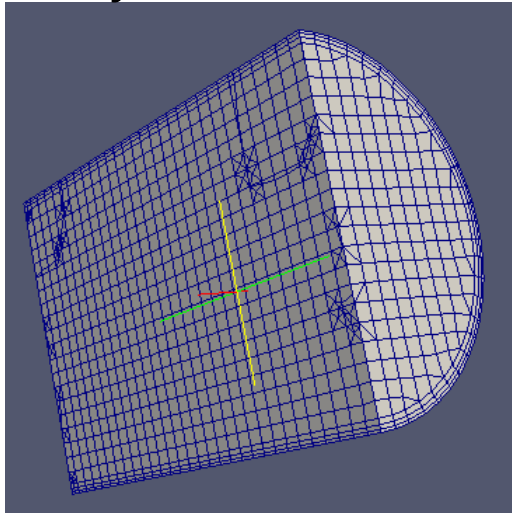
レイヤが追加  
されている

# でき上がったmeshにレイヤをつける方法

現在のmesh



layer追加



```
snappyHexMeshDict
7 \*-----
8 FoamFile
9 {
10     version 2.0;
11     format ascii;
12     class dictionary;
13     location system;
14     object snappyHexMeshDict;
15 }
16
17 castellatedMesh false;
18 snap false;
19 addLayers true;
20
21 geometry
22 {
23 }
24
25 castellatedMeshControls
26 {
27     features
28     (
29     );
30     refinementSurfaces
31     {
32     }
33     refinementRegions
34     {
35     }
36 }
```

## layerの設定

```
addLayersControls
{
    layers
    {
        sideWall ← patch名を指定
        {
            nSurfaceLayers 3;
        }
    }
    nSurfaceLayers 3;
    relativeSizes true;
    finalLayerThickness 0.3;
    minThickness 0.25;
    expansionRatio 1.4;
    nGrow 0;
    featureAngle 90;
    slipFeatureAngle 30;
    nRelaxIter 5;
    nSmoothSurfaceNormals 1;
    nSmoothNormals 3;
    nSmoothThickness 10;
    maxFaceThicknessRatio 0.5;
    maxThicknessToMedialRatio 0.3;
    minMedianAxisAngle 90;
    nBufferCellsNoExtrude 0;
    nLayerIter 50;
    nRelaxedIter 20;
}
```

Dict内のlayerのみ設定して実行すれば、既存のmeshにlayerが追加できる

# multiRegionモデルの流体部にレイヤを追加する方法

<multiRegionのcase>

```
regCase
0
constant
air
polyMesh
solid
polyMesh
system
```

<mesh作成用のcase>

```
meshCase
0
constant
polyMesh
system
```

流体部のmeshをコピー

このcase内でlayerを追加し、  
元に戻す

流体部にレイヤを追加することができる。  
meshのコピーは、TreeFoam上から容易にできる。

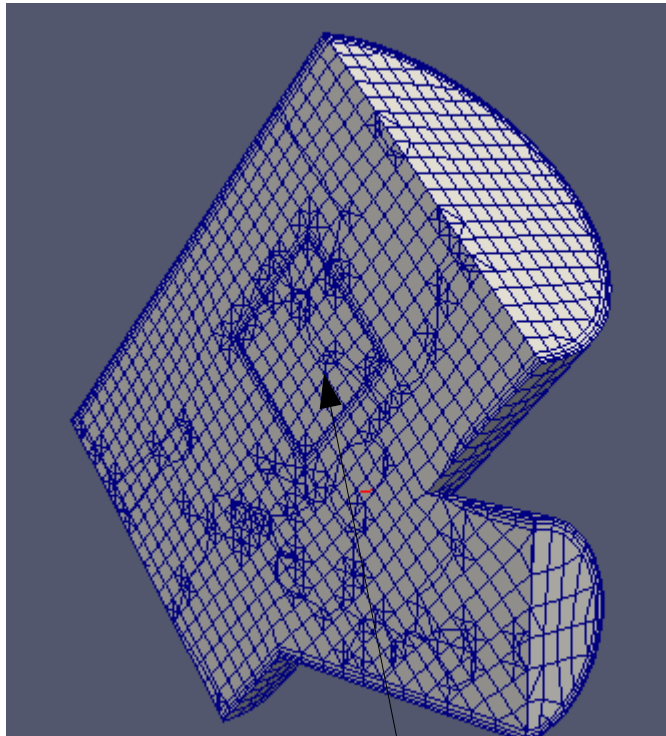


# snappyHexMeshによるlayerの追加について

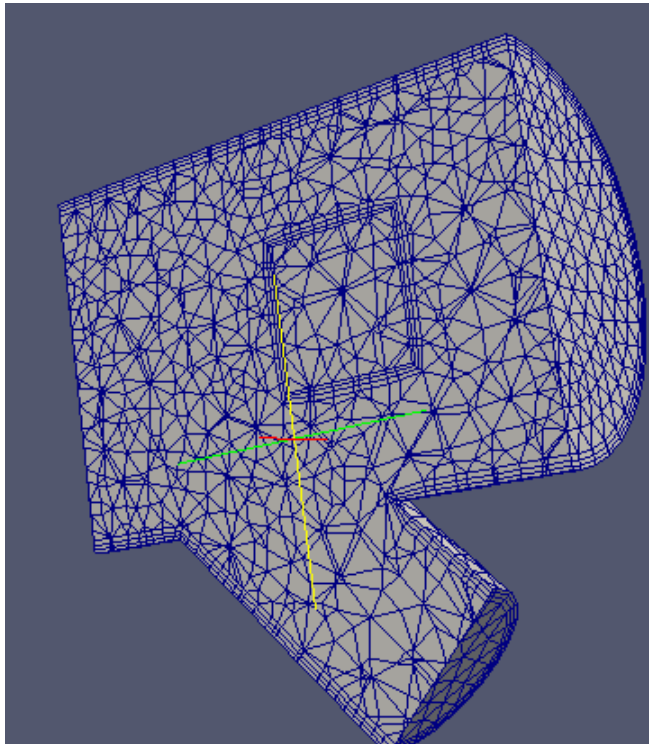
patch名を指定してlayerを追加する。

→ meshの種類(hexa、tetra、poly)でもlayerが追加できる。

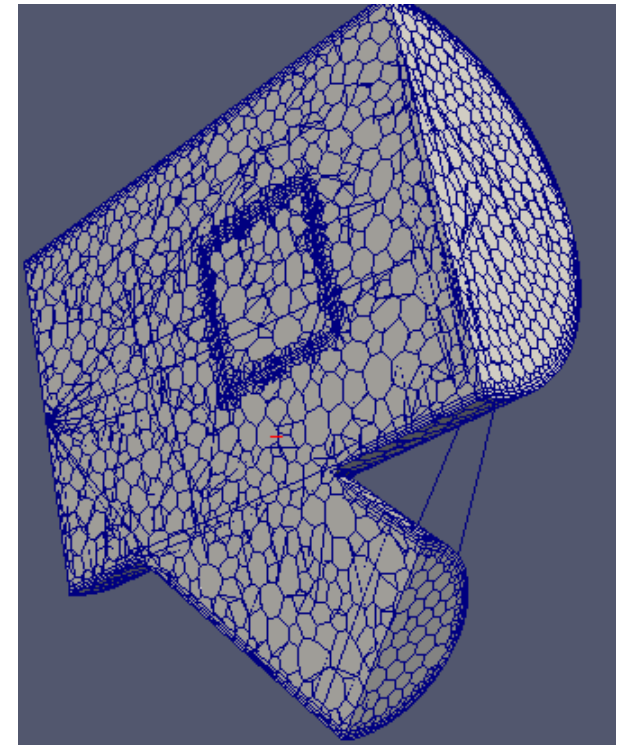
hexa



tetra



poly



固体領域



# TreeFoamにlayer作成機能を追加

