ただで始める流体解析

初心者が行う
プロペラチュートリアルの計算
（AMI，MRF）

Ver 2.1.x

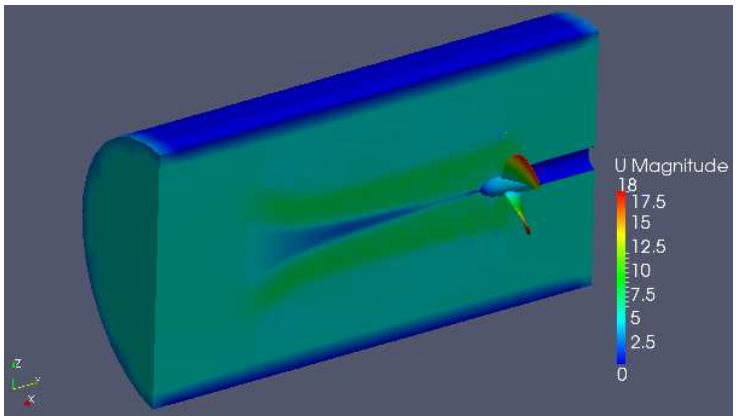
初心者が行うプロペラチュートリアルの計算

0．はじめに
1．OpenFOAMのチュートリアルを眺める
2．AMIの場合
　2-1：モデルの設定
　2-2：メッシュの作成
　2-3：計算の設定
　2-4：計算の結果
3．MRFの場合
　3-1：メッシュの作成
　3-2：計算の設定
　3-3：計算の結果,AMIとの比較
4．まとめ，その他

0. はじめに

0. はじめに

ファンの計算を今までMRFsimpleFoamの定常解析で実施してきましたが，OpenFOAMの正規版にAMI機能が入ってきたため，遅ればせながら勉強のためにプロペラのチュートリアルを色々と触ってみました。また，MRFとAMIを比較もしてみました。（今回のOpenFOAMのバージョンは2.1.xです）

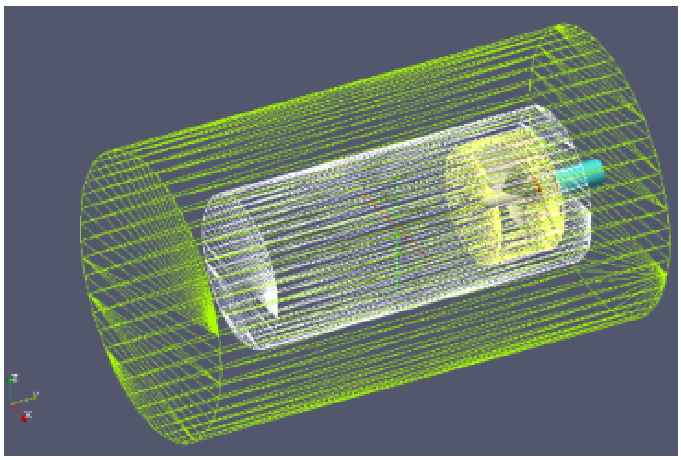


オープンCAE勉強会＠富山/20120929第4回の資料の中にも中川先生がアップされたプロペラチュートリアルの解説があります。

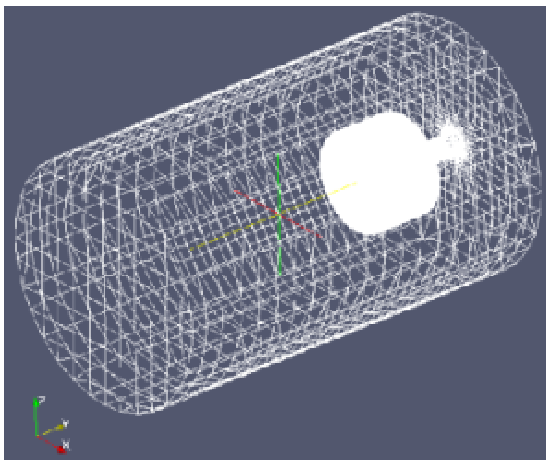
1．OpenFOAMの チュートリルを眺める
（手順を理解する）

1．OpenFOAMのチュートリアルを眺める
tutorials/incompressible/pimpleDyMFoam/propeller

●チュートリアルのモデル
Objファイル，メッシュを作るとこんな感じになっている。



Objファイル



メッシュ状態

Allrun.preの流れ

```
#!/bin/sh
cd ${0%/*} || exit 1    # run from this directory

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions
```

```
# copy propeller surface from resources folder
cp $FOAM_TUTORIALS/resources/geometry/propellerTip.obj.gz constant/triSurface/
```

①

```
# - meshing
```

```
runApplication blockMesh
```

②

```
surfaces="
    innerCylinder
    innerCylinderSmall
    outerCylinder
    propellerTip
    propellerStem1
    propellerStem2
    propellerStem3
"

for s in $surfaces; do
    runApplication surfaceFeatureExtract -includedAngle 150 -minElem 10       \
        constant/triSurface/$s.obj $s
    mv log.surfaceFeatureExtract log.surfaceFeatureExtract.$s
done
```

③

```
runApplication snappyHexMesh -overwrite
```

④

Allrun.preの流れ（つづき）

```
# - generate face/cell sets and zones
```

```
#runApplication setSet -batch removeRedundantZones.setSet
#mv log.setSet log.removeRedundantZones.setSet
runApplication topoSet -dict system/removeRedundantZones.topoSetDict
mv log.topoSet log.removeRedundantZones.topoSet
```

⑤

```
#runApplication setSet -batch createInletOutletSets.setSet
#mv log.setSet log.createInletOutletSets.setSet
runApplication topoSet -dict system/createInletOutletSets.topoSetDict
mv log.topoSet log.createInletOutletSets.topoSet
```

⑥

```
#runApplication setSet -batch createAMIFaces.setSet
#mv log.setSet log.createAMIFaces.setSet
runApplication topoSet -dict system/createAMIFaces.topoSetDict
mv log.topoSet log.createAMIFaces.topoSet
```

⑦

```
# - create the inlet/outlet patches
runApplication createPatch -overwrite
```

⑧

```
# force removal of fields generated by snappy
\rm -rf 0
```

```
createBaffles -internalFacesOnly -overwrite innerCylinderSmall '(AMI1 AMI2)'   \
    > log.createBaffles 2>&1
```

⑨

```
runApplication mergeOrSplitBaffles -split -overwrite
```

⑩

```
# - apply the initial fields

cp -rf 0.org 0
```

●Allrun.preでの基本的なメッシュ作成の流れを順次見て行く

①チュートリアルから形状ファイルをコピー

```
# copy propeller surface from resources folder
cp $FOAM_TUTORIALS/resources/geometry/propellerTip.obj.gz constant/triSurface/
```

②blockMesh　　基礎メッシュの作成

```
runApplication blockMesh
```

③surfaceFeatureExtract　　特徴線の取出し

（Ver2.2.xでは設定書式が異なっている）

```
surfaces="
    innerCylinder
    innerCylinderSmall
    outerCylinder
    propellerTip
    propellerStem1
    propellerStem2
    propellerStem3
"

for s in $surfaces; do
    runApplication surfaceFeatureExtract -includedAngle 150 -minElem 10        \
        constant/triSurface/$s.obj $s
    mv log.surfaceFeatureExtract log.surfaceFeatureExtract.$s
done
```


④snappyHexMesh　　全体メッシュの作成

```
runApplication snappyHexMesh -overwrite
```

```
castellatedMesh true;
snap            true;
addLayers       false;
```

境界層の設定はなし

形状ファイルの設定

```
geometry
{
    innerCylinder.obj
    {
        type        triSurfaceMesh;
        name        innerCylinder;
        regions
        {
            ascii
            {
                name        innerCylinder;
            }
        }
    }
    innerCylinderSmall.obj
    {
        type        triSurfaceMesh;
        name        innerCylinderSmall;
        regions
        {
            ascii
            {
                name        innerCylinderSmall;
            }
        }
    }
```

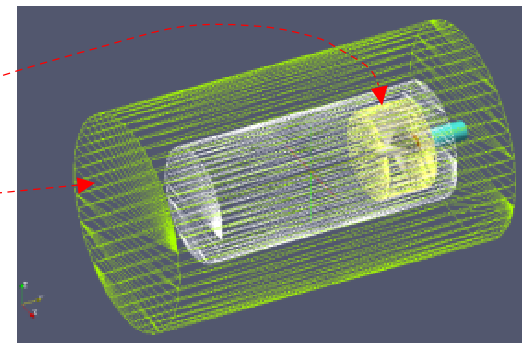
```
# Wavefront OBJ file
# Regions:
#     0    ascii
#
# points    : 100
# triangles : 196
#
v 0.0857323 0.1 -0.135092
v -0.0857323 0.1 -0.135092
v -0.0857323 0.1 0.135092
v 0.0857323 0.1 0.135092
v 0.158738 0.1 -0.0200533
v -0.158738 0.1 -0.0200533
v -0.158738 0.1 0.0200533
v 0.158738 0.1 0.0200533
```

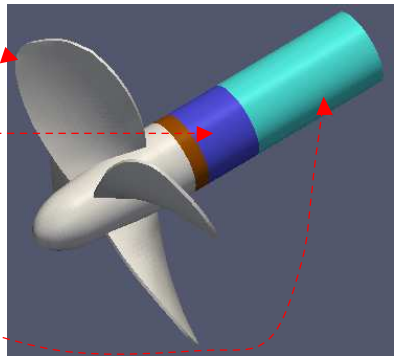
形状はWavefront社のAdvanced Visualizerというソフト用のファイルフォーマットとなっている

④snappyHexMesh　　全体メッシュの作成（つづき）

フィチャーエッジの設定

```
features
(
    {
        file        "innerCylinderSmall.eMesh";
        level       4;
    }
    {
        file        "outerCylinder.eMesh";
        level       0;
    }
    {
        file        "propellerTip.eMesh";
        level       4;
    }
      {
          file        "propellerStem1.eMesh";
          level       4;
      }
    {
        file        "propellerStem2.eMesh";
        level       4;
    }
    {
        file        "propellerStem3.eMesh";
        level       4;
    }
);
```

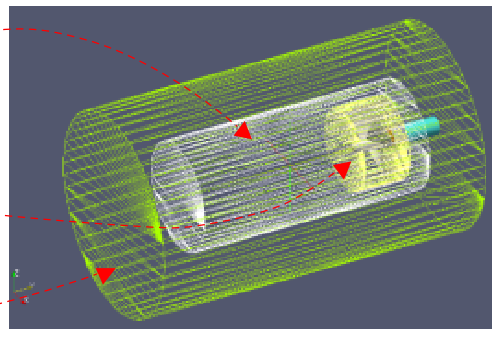


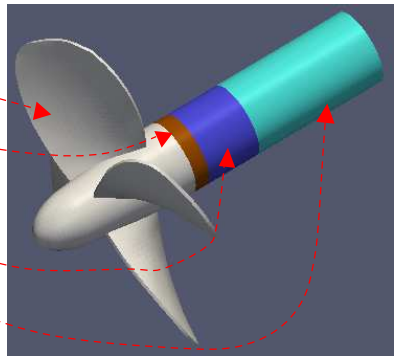


④snappyHexMesh　　全体メッシュの作成（つづき）

表面のリファイン設定

```
refinementSurfaces
{
    innerCylinder
    {
        level       (2 3);
        cellZone    innerCylinder;
        faceZone    innerCylinder;
        cellZoneInside  inside;
    }
    innerCylinderSmall
    {
        level       (4 4);
        cellZone    innerCylinderSmall;
        faceZone    innerCylinderSmall;
        cellZoneInside  inside;
    }
    outerCylinder
    {
        level       (0 0);
    }
    propellerTip
    {
        level       (4 5);
    }
    propellerStem1
    {
        level       (4 4);
    }
    propellerStem2
    {
        level       (4 4);
    }
    propellerStem3
    {
        level       (4 4);
    }
}
```

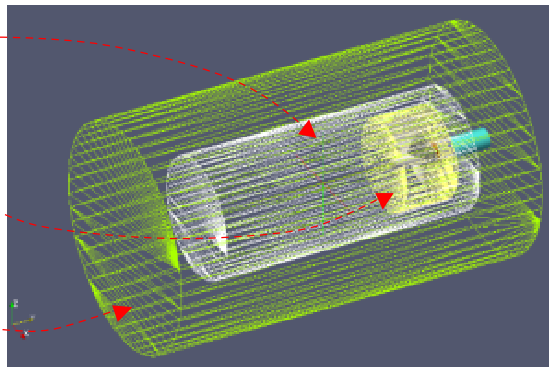




④snappyHexMesh　　全体メッシュの作成（つづき）

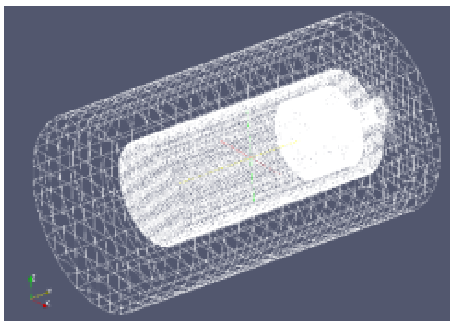
ボディーのリファイン設定

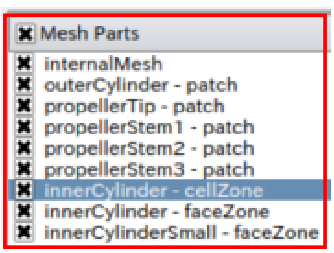
```
refinementRegions
{
    innerCylinder
    {
        mode        inside;
        levels      ((1E15 3));
    }
    innerCylinderSmall
    {
        mode        inside;
        levels      ((1E15 4));
    }
    outerCylinder
    {
        mode        inside;
        levels      ((1E15 0));
    }
}
```

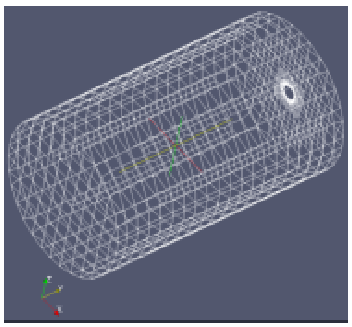


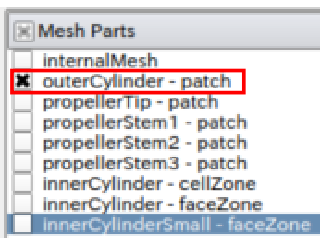
④snappyHexMesh　　全体メッシュの作成（つづき）

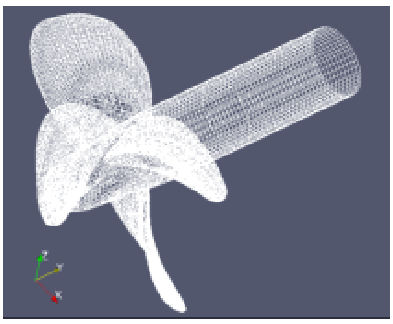
作成されたメッシュを確認する

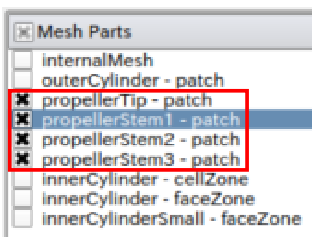




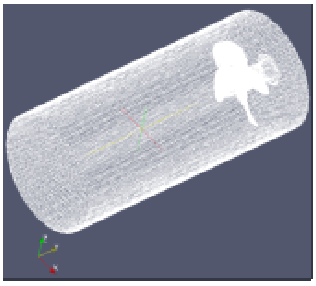
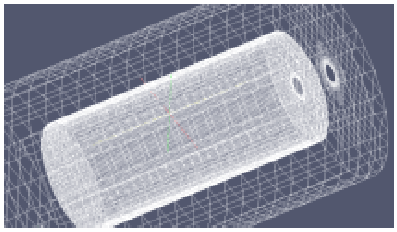
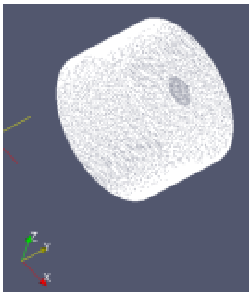
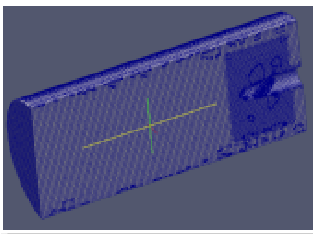
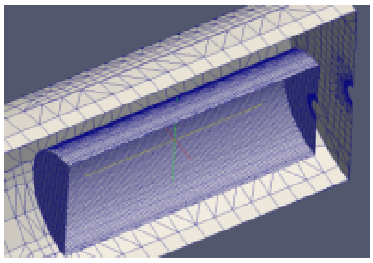
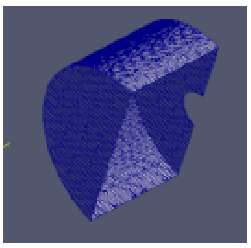


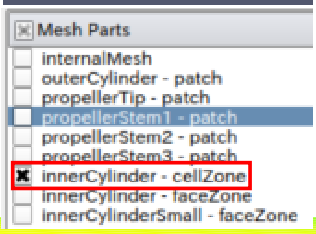


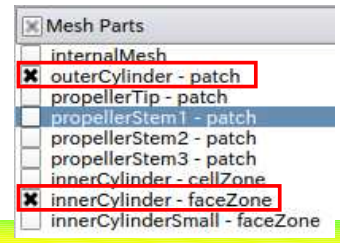


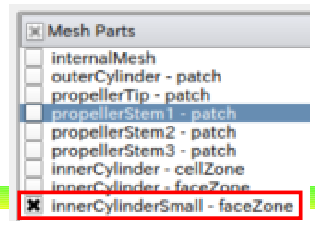


④snappyHexMesh　　全体メッシュの作成（つづき）







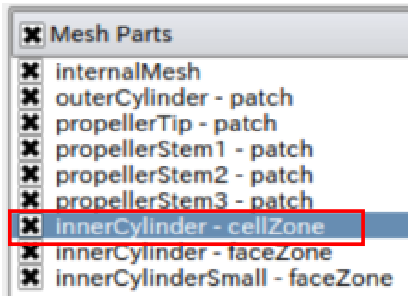
④snappyHexMesh　　全体メッシュの作成（つづき）

constant/polyMeshの下にある
cellZonesのファイルをエディタで開く

```
FoamFile
{
    version     2.0;
    format      binary;
    class       regIOobject;
    location    "constant/polyMesh";
    object      cellZones;
}
// * * * * * * * * * * * * * * * * * *

2
(
innerCylinder
{
    type cellZone;
cellLabels      List<label>
491182
```

```
innerCylinderSmall
{
    type cellZone;
cellLabels
0
;
}
)
```

セル数は0のため
paraFoamでは表示されない

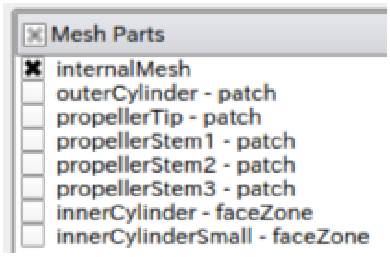


●ここからがAMI特有の設定になる

⑤snappyHexMeshで作成されたcellZoneの消去

```
runApplication topoSet -dict system/removeRedundantZones.topoSetDict
mv log.topoSet log.removeRedundantZones.topoSet
```

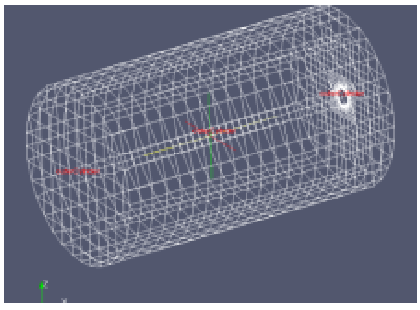
```
{
    name    innerCylinder;
    type    cellZoneSet;
    action  remove;
}
{
    name    innerCylinderSmall;
    type    cellZoneSet;
    action  remove;
}
```

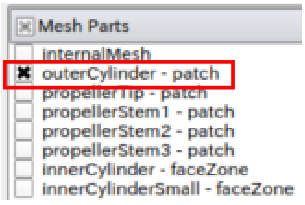


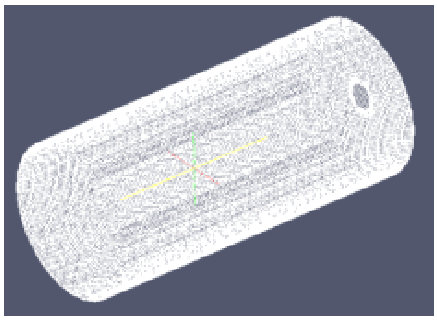
cellZoneが消去される

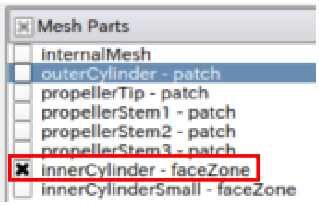
⑤snappyHexMeshで作成されたcellZoneの消去（つづき）

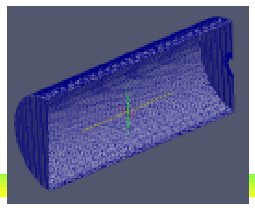
メッシュ状態のチェック

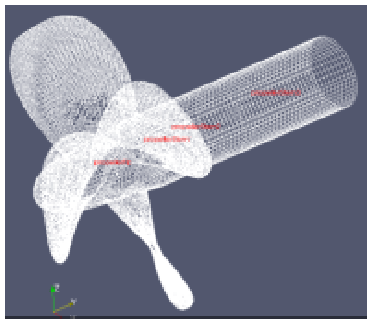


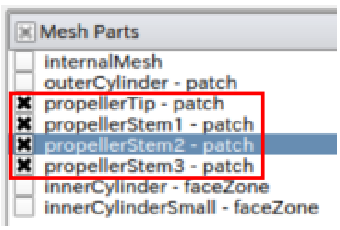




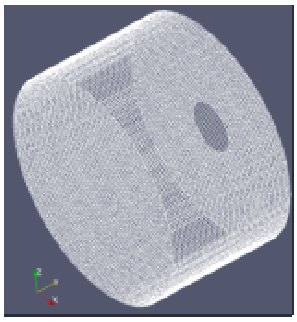


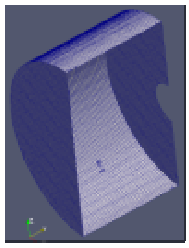


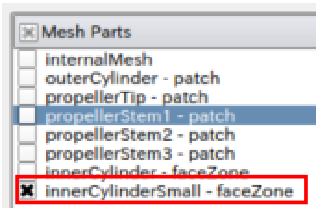




⑤snappyHexMeshで作成されたcellZoneの消去（つづき）
メッシュ状態のチェック





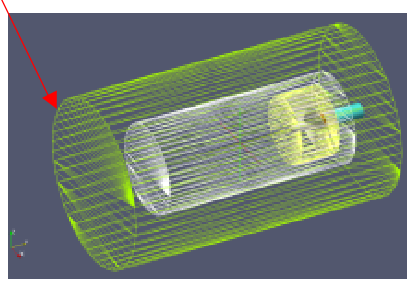


⑥インレットとアウトレットのパッチの作成

```
runApplication topoSet -dict system/createInletOutletSets.topoSetDict
mv log.topoSet log.createInletOutletSets.topoSet
```

```
{
    name    boundaryFaces;
    type    faceSet;
    action  new;
    source  patchToFace;
    sourceInfo
    {
        name outerCylinder;
    }
}

{
    name    outletFaces;
    type    faceSet;
    action  new;
    source  faceToFace;
    sourceInfo
    {
        set boundaryFaces;
    }
}
```

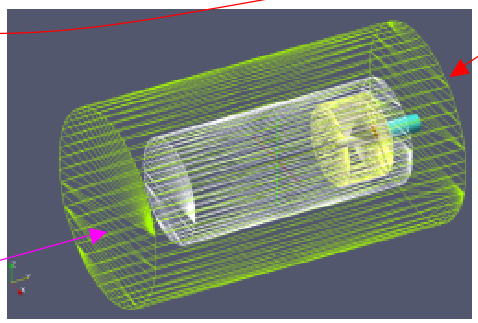
```
{
    name    inletFaces;
    type    faceSet;
    action  new;
    source  faceToFace;
    sourceInfo
    {
        set boundaryFaces;
    }
}
```

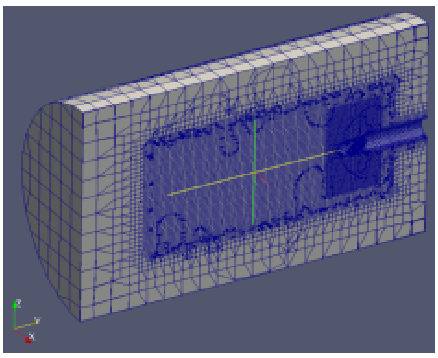


boundaryFaces
outletFaces
inletFaces
ともに同じ面情報が入る

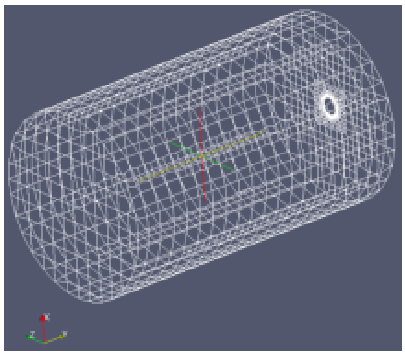
⑥インレットとアウトレットのパッチの作成（つづき）

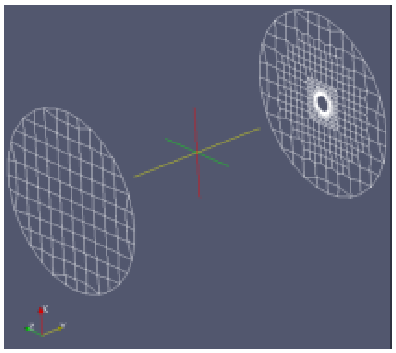
```
{
    name    outletFaces;
    type    faceSet;
    action  subset;
    source  normalToFace;
    sourceInfo
    {
        normal  (0 -1 0);    // Vector
        cos     0.3;         // Tolerance (max cos of angle)
    }
}

{
    name    inletFaces;
    type    faceSet;
    action  subset;
    source  normalToFace;
    sourceInfo
    {
        normal  (0 1 0);     // Vector
        cos     0.3;         // Tolerance (max cos of angle)
    }
}
```

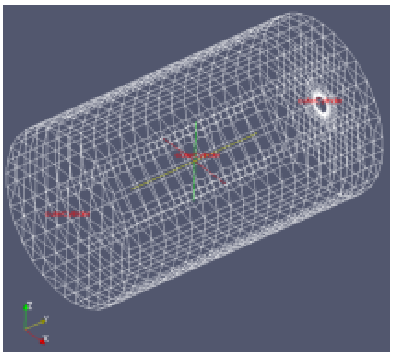


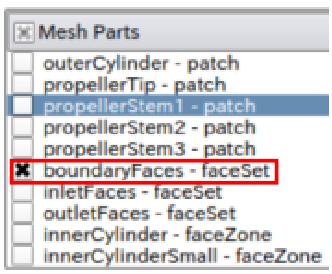


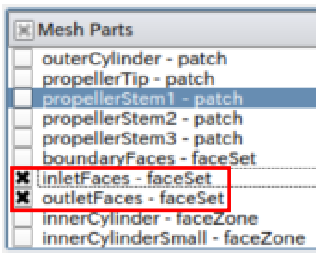
⑥インレットとアウトレットのパッチの作成（つづき）
メッシュ状態のチェック

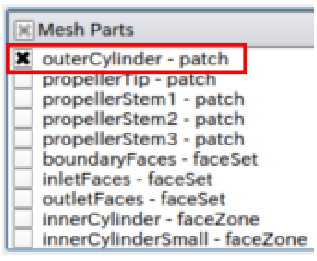




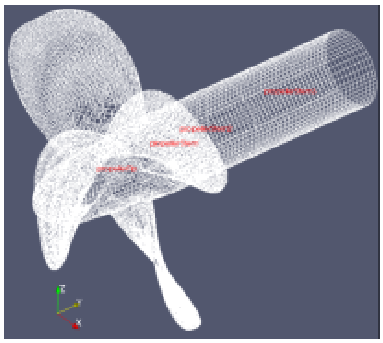


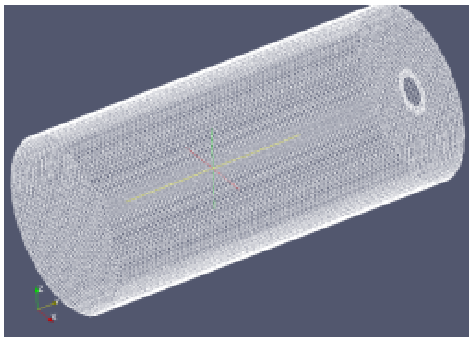


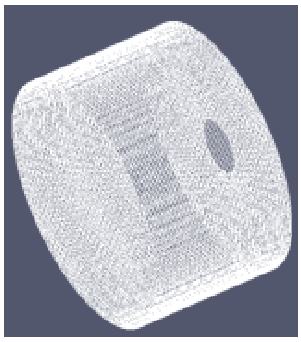


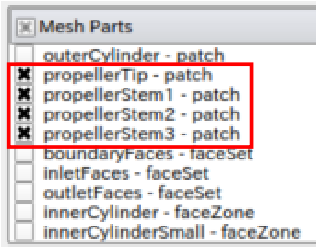


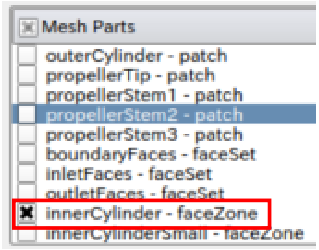
⑥インレットとアウトレットのパッチの作成（つづき）
メッシュ状態のチェック

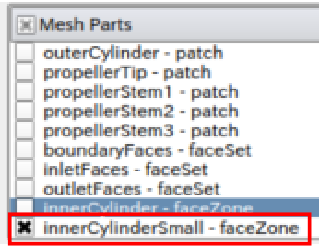












⑦回転領域の作成

```
runApplication topoSet -dict system/createAMIFaces.topoSetDict
mv log.topoSet log.createAMIFaces.topoSet
```

この部分は分けて実施してどうなっているか見て行く（今回は6分割）

7-1
```
{
    name    innerCylinderSmall;
    type    cellSet;
    action  new;
    source  cylinderToCell;
    sourceInfo
    {
        p1      (0 -0.08 0);
        p2      (0 0.06 0);
        radius  0.12;
    }
}
{
    name    outerCells;
    type    cellSet;
    action  new;
    source  cellToCell;
    sourceInfo
    {
        set innerCylinderSmall;
    }
}
{
    name    outerCells;
    type    cellSet;
    action  invert;
}
```

7-2
```
{
    name    innerCylinderSmall;
    type    cellZoneSet;
    action  new;
    source  setToCellZone;
    sourceInfo
    {
        set     innerCylinderSmall;
    }
}
```

7-3
```
{
    name    innerCylinderSmallFace;
    type    faceSet;
    action  new;
    source  cellToFace;
    sourceInfo
    {
        set     innerCylinderSmall;
        option  all;
    }
}
```

7-4
```
{
    name    innerCylinderSmallFace;
    type    faceSet;
    action  subset;
    source  cellToFace;
    sourceInfo
    {
        set     outerCells;
        option  all;
    }
}
```

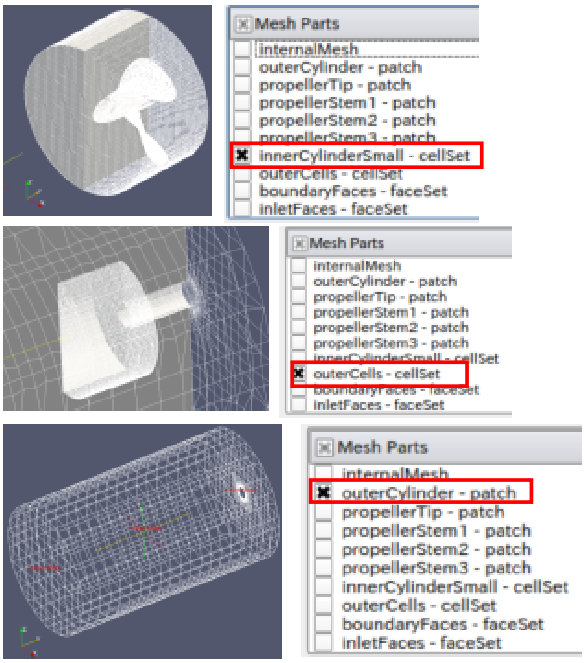
7-5
```
{
    name    innerCylinderSmall;
    type    faceZoneSet;
    action  new;
    source  setsToFaceZone;
    sourceInfo
    {
        faceSet     innerCylinderSmallFace;
        cellSet     innerCylinderSmall;
    }
}
```

7-6
```
// Dummy faceSet for creating initial patches
{
    name    dummyFaces;
    type    faceSet;
    action  new;
    source  labelToFace;
    sourceInfo
    {
        value ();
    }
}
```

⑦回転領域の作成（つづき）

```
{
    name    innerCylinderSmall;
    type    cellSet;
    action  new;
    source  cylinderToCell;
    sourceInfo
    {
        p1      (0 -0.08 0);
        p2      (0 0.06 0);
        radius  0.12;
    }
}
{
    name    outerCells;
    type    cellSet;
    action  new;
    source  cellToCell;
    sourceInfo
    {
        set innerCylinderSmall;
    }
}
{
    name    outerCells;
    type    cellSet;
    action  invert;
}
```

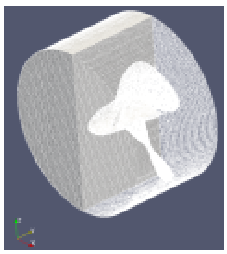
7-1

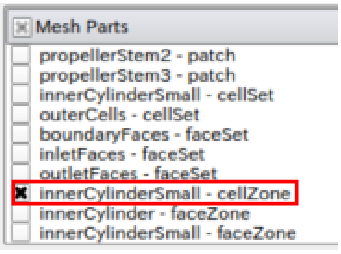


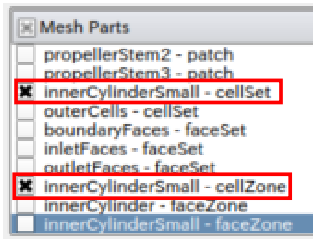
⑦回転領域の作成（つづき）

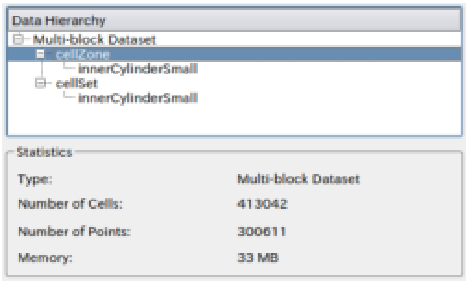
```
{
    name    innerCylinderSmall;
    type    cellZoneSet;
    action  new;
    source  setToCellZone;
    sourceInfo
    {
        set     innerCylinderSmall;
    }
}
```

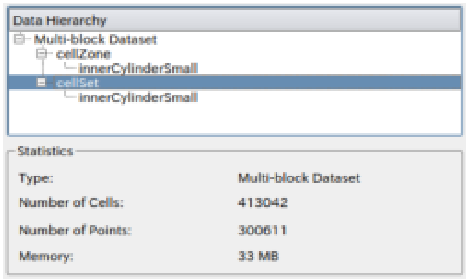
7-2











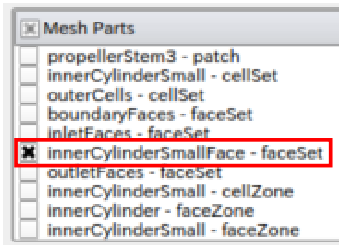
cellZone，cellSetは同じものが入っている？

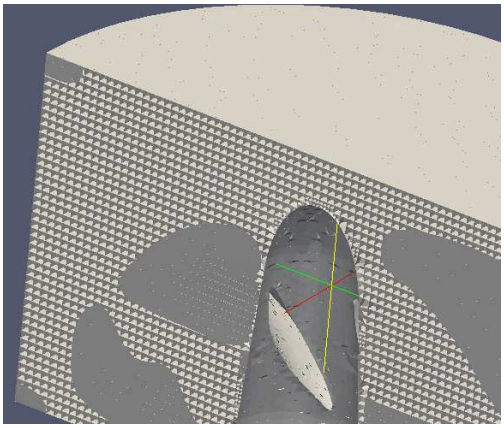
http://www.cfd-online.com/Forums/openfoam/90839-cellsets-vs-cellzones.htmlにcellZoneとcellSetの差の説明がある

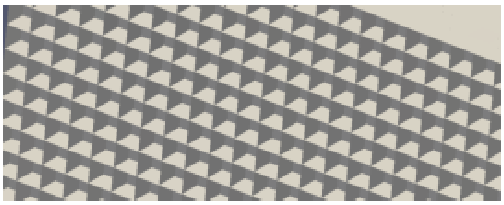
⑦回転領域の作成（つづき）

```
{
    name    innerCylinderSmallFace;
    type    faceSet;
    action  new;
    source  cellToFace;
    sourceInfo
    {
        set     innerCylinderSmall;
        option  all;
    }
}
```

7-3

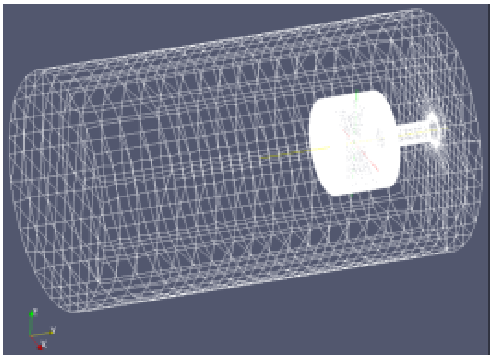


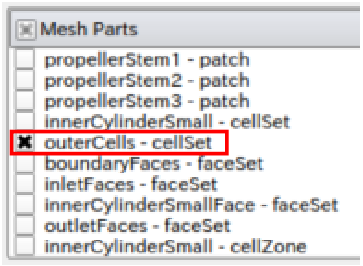


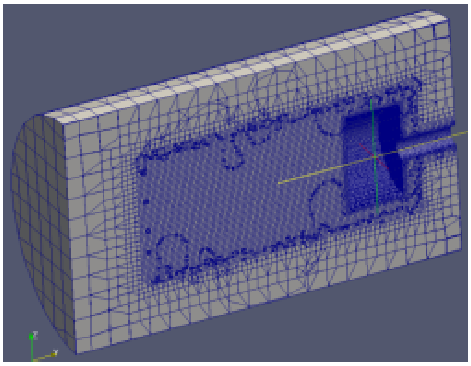


innerCylinderSamllで作成されたcellを構成する全てのfaceが抜き出される。
表面だけでなく内部の面も抽出される。

⑦回転領域の作成（つづき）



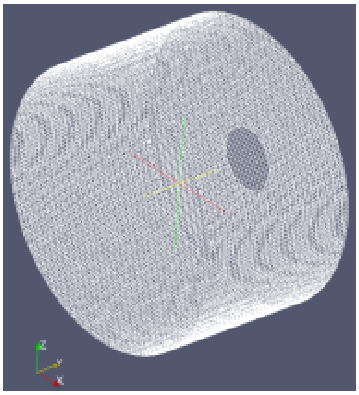


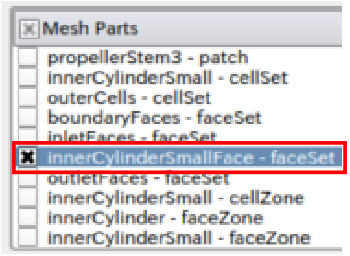


⑦回転領域の作成（つづき）

```
{
    name    innerCylinderSmallFace;
    type    faceSet;
    action  subset;
    source  cellToFace;
    sourceInfo
    {
        set     outerCells;
        option  all;
    }
}
```

7-4

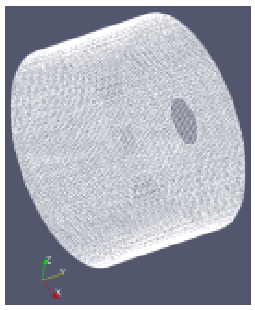


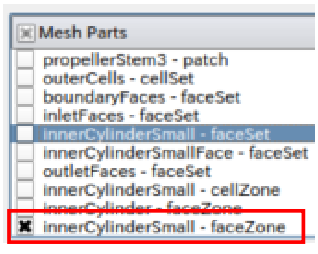


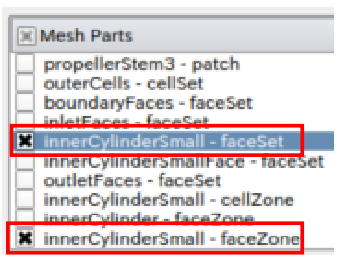
⑦回転領域の作成（つづき）

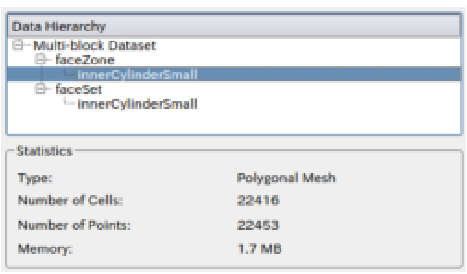
```
{
    name    innerCylinderSmall;
    type    faceZoneSet;
    action  new;
    source  setsToFaceZone;
    sourceInfo
    {
        faceSet     innerCylinderSmallFace;
        cellSet     innerCylinderSmall;
    }
}
```

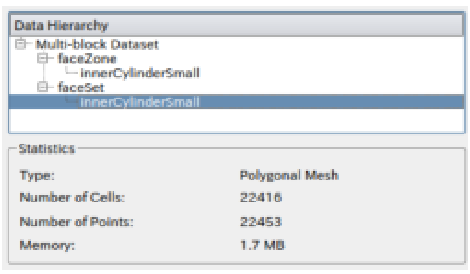
7-5











faseZone，faceSetは同じものが入っている？

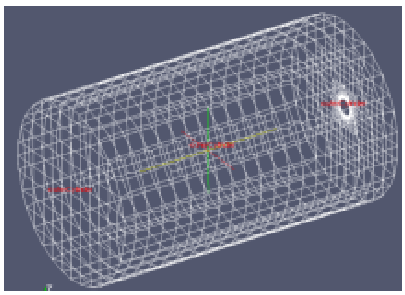
⑦回転領域の作成（つづき）

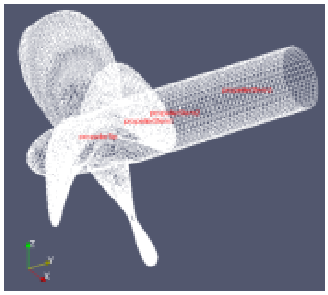
```
// Dummy faceSet for creating initial patches
{
    name    dummyFaces;
    type    faceSet;
    action  new;
    source  labelToFace;
    sourceInfo
    {
        value ();
    }
}
```

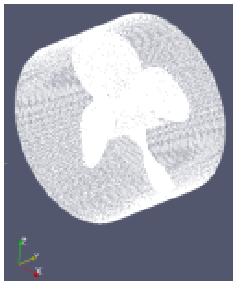
7-6

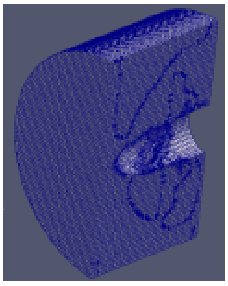
dummyFacesには何も設定されない

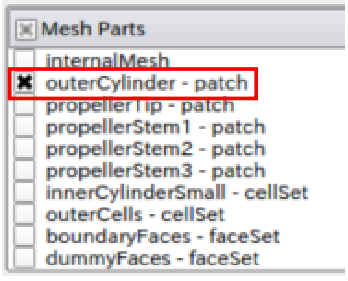
⑦回転領域の作成（つづき） ここまでのメッシュ状態の確認してみる

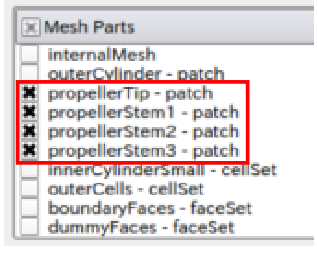


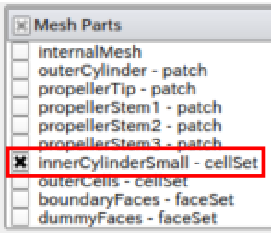




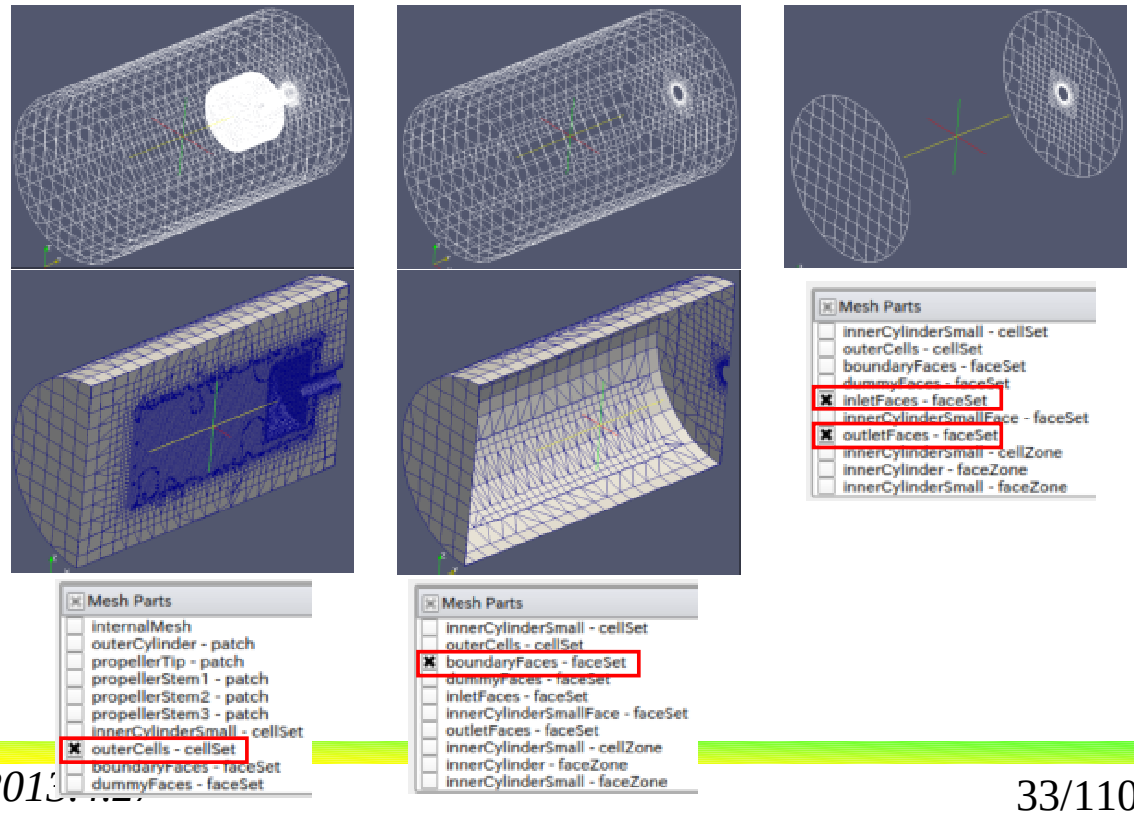




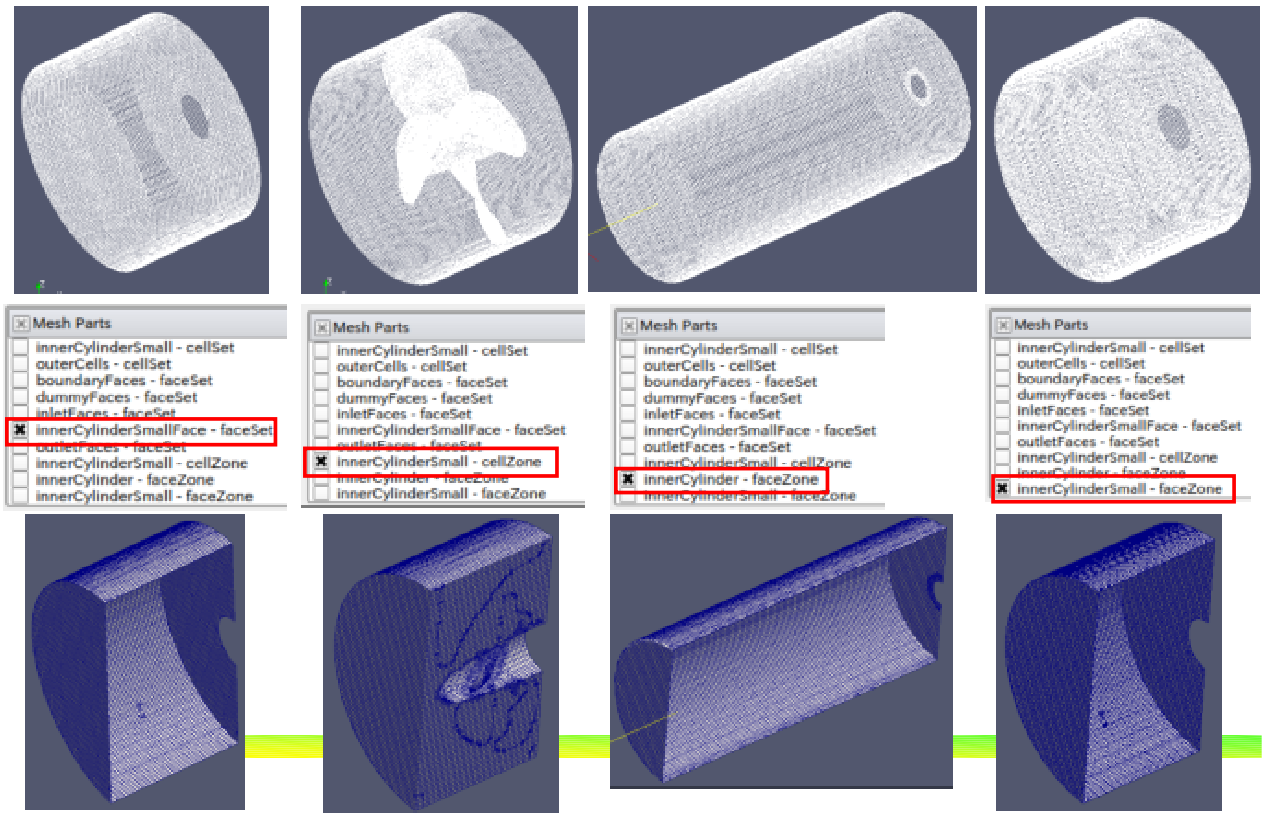




⑦回転領域の作成（つづき） ここまでのメッシュ状態の確認してみる



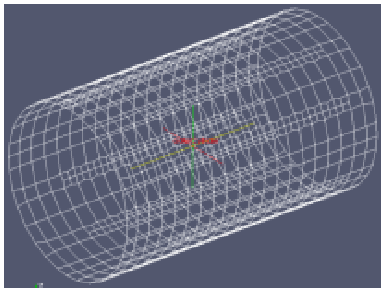
⑦回転領域の作成（つづき） ここまでのメッシュ状態の確認してみる

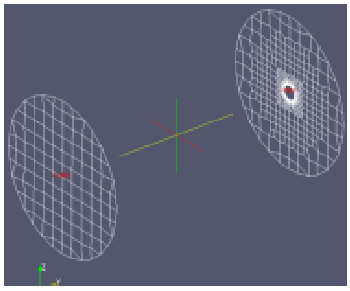


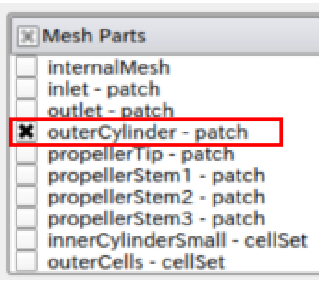
⑧パッチの作成　インレット，アウトレット

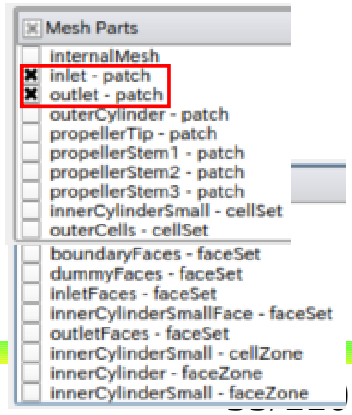
```
# - create the inlet/outlet patches
runApplication createPatch -overwrite
```

```
{
    name inlet;

    patchInfo
    {
        type            patch;
    }

    constructFrom set;

    set inletFaces;
}
{
    name outlet;

    patchInfo
    {
        type            patch;
    }

    constructFrom set;

    set outletFaces;
}
```









outerCylinderの左右の
パッチが削除されている

⑧パッチの作成　回転境界のダミー

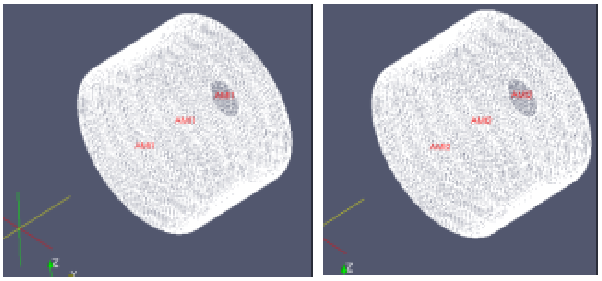
```
// Construct empty patches
{
    name AMI1;
    patchInfo
    {
        type            cyclicAMI;
        neighbourPatch  AMI2;
        transform       noOrdering;
        surface
        {}
    }
    constructFrom set;
    set dummyFaces;
}
{
    name AMI2;

    patchInfo
    {
        type            cyclicAMI;
        neighbourPatch  AMI1;
        transform       noOrdering;
        surface
        {}
    }
    constructFrom set;
    set dummyFaces;
}
```

回転領域を作成するため
のダミーフェースを設定？

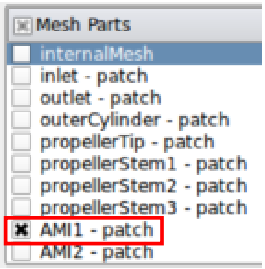
⑨回転境界の作成

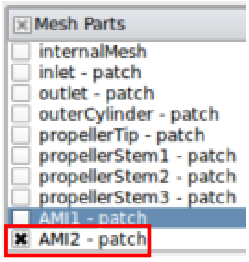
```
createBaffles -internalFacesOnly -overwrite innerCylinderSmall '(AMI1 AMI2)'   \
    > log.createBaffles 2>&1
```

AMI1，AMI2の回転領域を作成



```
Usage: createBaffles [OPTIONS] <faceZone> <(masterPatch slavePatch)>
options:
  -additionalPatches <((master2 slave2) .. (masterN slaveN))>
  -case <dir>       specify alternate case directory, default is the cwd
  -internalFacesOnly
                    do not convert boundary faces
  -noFunctionObjects
                    do not execute functionObjects
  -overwrite        overwrite existing mesh/results files
  -parallel         run in parallel
  -region <name>    specify alternative mesh region
  -roots <(dir1 .. dirN)>
                    slave root directories for distributed running
  -updateFields     update fields to include new patches: NOTE: updated field
                    values may need to be edited
  -srcDoc           display source code in browser
  -doc              display application documentation in browser
  -help             print the usage

Makes internal faces into boundary faces.
Does not duplicate points, unlike mergeOrSplitBaffles.
```

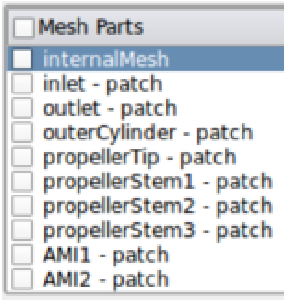




⑩回転境界の分離

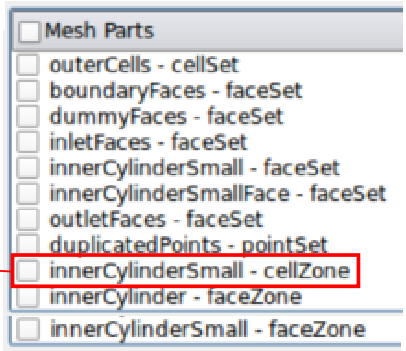
```
runApplication mergeOrSplitBaffles -split -overwrite
```

```
Usage: mergeOrSplitBaffles [OPTIONS]
options:
  -case <dir>       specify alternate case directory, default is the cwd
  -detectOnly       find baffles only, but do not merge or split them
  -noFunctionObjects
                    do not execute functionObjects
  -overwrite        overwrite existing mesh/results files
  -parallel         run in parallel
  -region <name>    specify alternative mesh region
  -roots <(dir1 .. dirN)>
                    slave root directories for distributed running
  -split            topologically split duplicate surfaces
  -srcDoc           display source code in browser
  -doc              display application documentation in browser
  -help             print the usage

Detect faces that share points (baffles).
Merge them or duplicate the points.
```

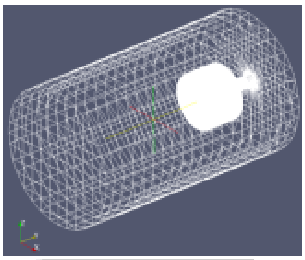
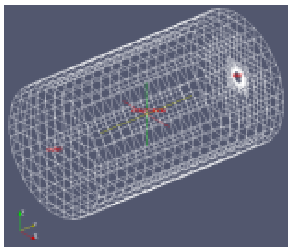
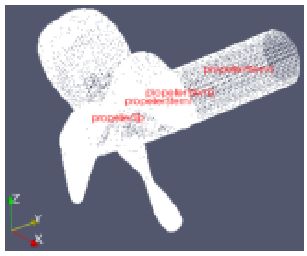


回転させるセル

```
dynamicFvMesh   solidBodyMotionFvMesh;

motionSolverLibs ( "libfvMotionSolvers.so" );

solidBodyMotionFvMeshCoeffs
{
    cellZone        innerCylinderSmall;

    solidBodyMotionFunction  rotatingMotion;
    rotatingMotionCoeffs
    {
        CofG        (0 0 0);
        radialVelocity (0 9000 0); // deg/s
    }
}
```



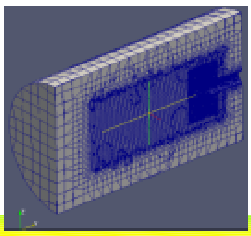
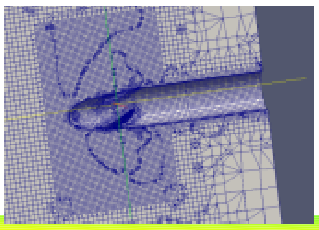
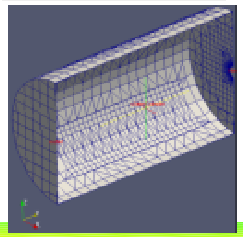
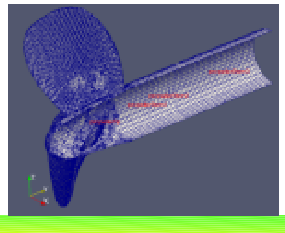
⑩回転境界の分離（つづき）
最終的なメッシュの作成状態を確認する。

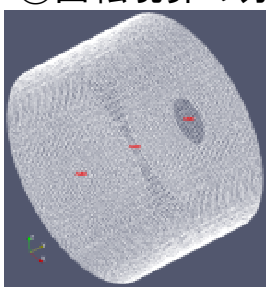
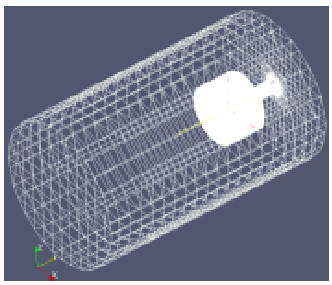
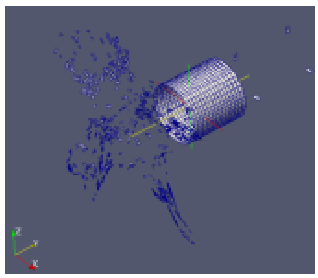
Mesh Parts
- internalMesh
- inlet - patch
- outlet - patch
- outerCylinder - patch
- propellerTip - patch
- propellerStem1 - patch
- propellerStem2 - patch
- propellerStem3 - patch
- AMI1 - patch
- AMI2 - patch

Mesh Parts
- internalMesh
- inlet - patch
- outlet - patch
- outerCylinder - patch
- propellerTip - patch
- propellerStem1 - patch
- propellerStem2 - patch
- propellerStem3 - patch
- AMI1 - patch
- AMI2 - patch

Mesh Parts
- internalMesh
- inlet - patch
- outlet - patch
- outerCylinder - patch
- propellerTip - patch
- propellerStem1 - patch
- propellerStem2 - patch
- propellerStem3 - patch
- AMI1 - patch
- AMI2 - patch

⑩回転境界の分離（つづき）

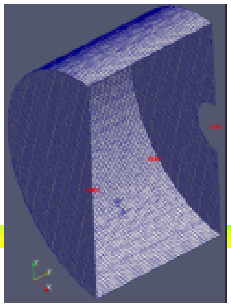
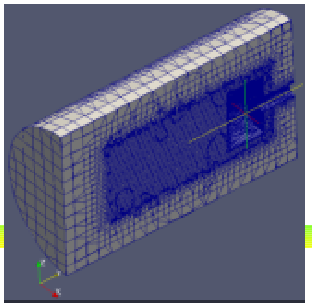
表示する要素なし

Mesh Parts
- outlet - patch
- outerCylinder - patch
- propellerTip - patch
- propellerStem1 - patch
- propellerStem2 - patch
- propellerStem3 - patch
- AMI1 - patch
- AMI2 - patch
- outerCells - cellSet
- boundaryFaces - faceSet

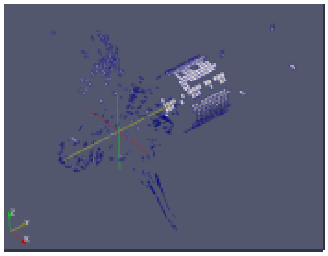
Mesh Parts
- propellerStem1 - patch
- propellerStem2 - patch
- propellerStem3 - patch
- AMI1 - patch
- AMI2 - patch
- outerCells - cellSet
- boundaryFaces - faceSet
- dummyFaces - faceSet
- inletFaces - faceSet
- innerCylinderSmall - faceSet

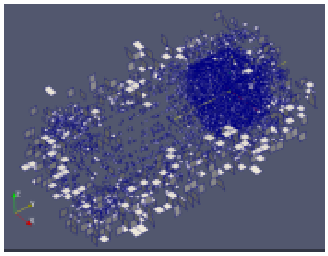
Mesh Parts
- AMI1 - patch
- AMI2 - patch
- outerCells - cellSet
- boundaryFaces - faceSet
- dummyFaces - faceSet
- inletFaces - faceSet
- innerCylinderSmall - faceSet
- innerCylinderSmallFace - faceSet
- outletFaces - faceSet
- duplicatedPoints - pointSet

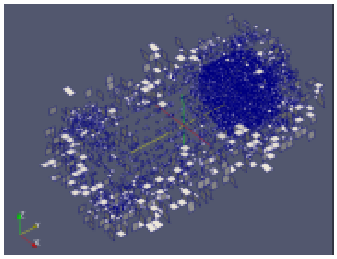
Mesh Parts
- AMI1 - patch
- AMI2 - patch
- outerCells - cellSet
- boundaryFaces - faceSet
- dummyFaces - faceSet
- inletFaces - faceSet
- innerCylinderSmall - faceSet
- innerCylinderSmallFace - faceSet
- outletFaces - faceSet
- duplicatedPoints - pointSet

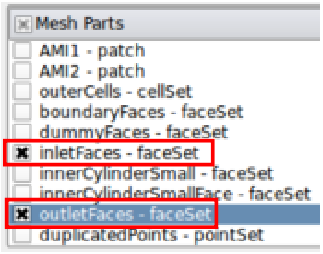
⑩回転境界の分離（つづき）

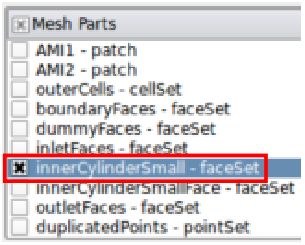


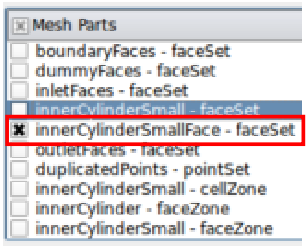


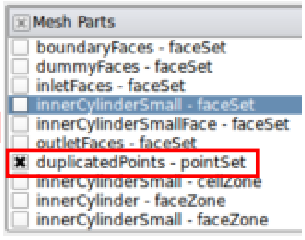


表示する要素なし

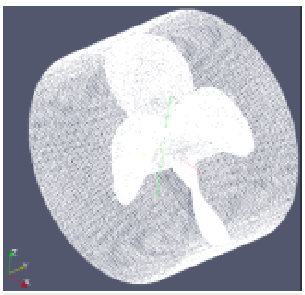


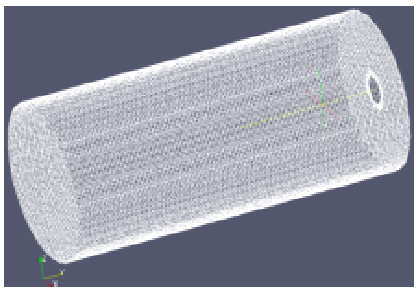


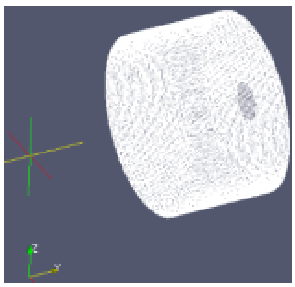


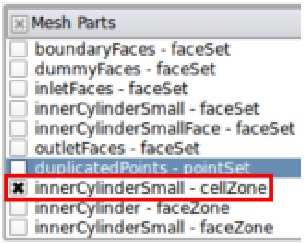


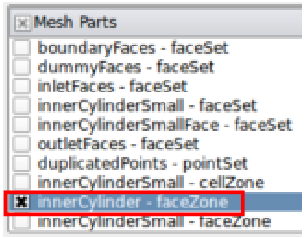
⑩回転境界の分離（つづき）

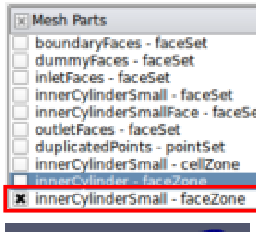










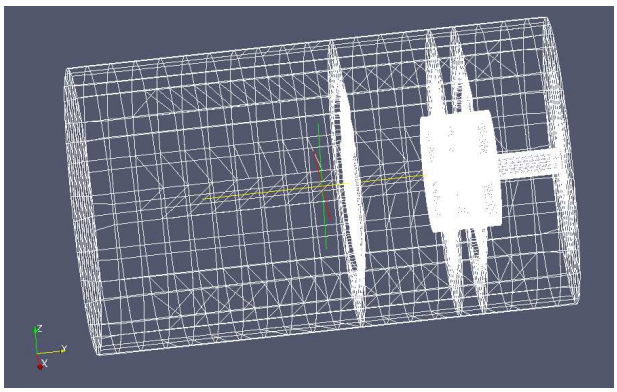


⑪計算の実施

```
runApplication decomposePar

runParallel `getApplication` 4

runApplication reconstructPar
```

計算は4並列で実施

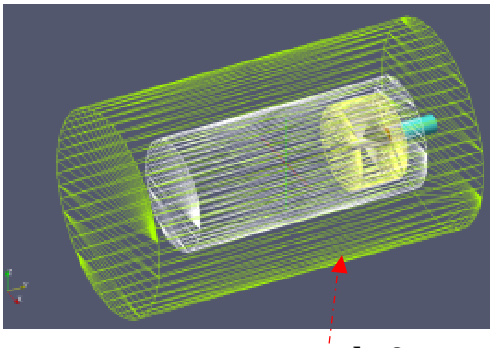
```
numberOfSubdomains 4;

method          hierarchical;

hierarchicalCoeffs
{
    n               (1 4 1);
    delta           0.001;
    order           xyz;
}
```



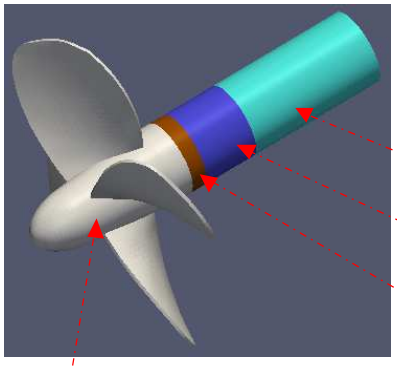
⑫計算結果のチェック

最終時間のy+を確認する。境界層がないため大きく出ている。

```
Patch 0 named outerCylinder_Mesh y+ : min: 1429 max: 3961.72 average: 2136.61

Patch 3 named propellerTip_PROPELLERTIP y+ : min: 27.8318 max: 967.274 average: 212.768

Patch 4 named propellerStem1_PROPELLERSTEM1 y+ : min: 136.948 max: 307.315 average: 218.785

Patch 5 named propellerStem2_PROPELLERSTEM2 y+ : min: 166.94 max: 322.628 average: 243.11

Patch 6 named propellerStem3_PROPELLERSTEM3 y+ : min: 102.974 max: 343.828 average: 215.198
```



Patch 0



Patch 6

Patch 5

Patch 4

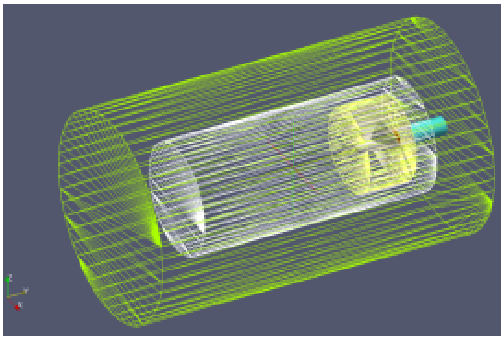
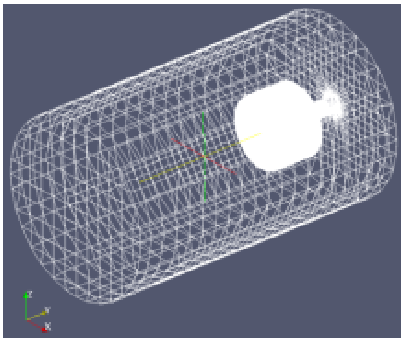
Patch 3

2．AMIの場合
（pimpleDyMFoam）

2-1．モデルの設定

・自分流のモデル構築方法に変更する。
・チュートリアルの形状は比較的簡単なため，topoSetを使って色々cellZone，cellSet，faceZone，faceSetを操作するが，実際の形状ではやり難い。
・チュートリアルのobjファイルの寸法を参考にしてstlファイルを作成する。
（出来るだけstlファイルを利用して設定する。上手く設定する事でtopoSet等の手間を軽減する。）

●チュートリアルのモデル
Objファイル，メッシュを見るとこんな感じになっている。（前述）

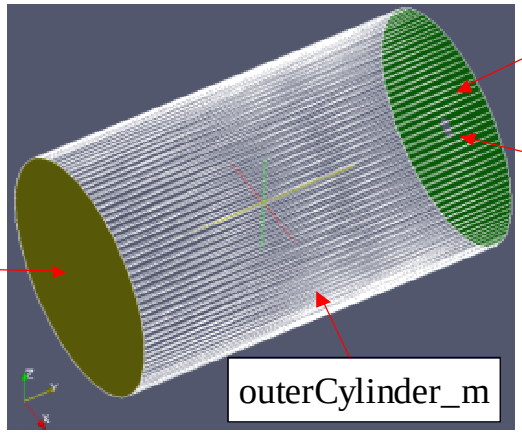



2-1．モデルの設定

●外部領域部分

チュートリアルでは1パートになっているものを，3つに分ける。

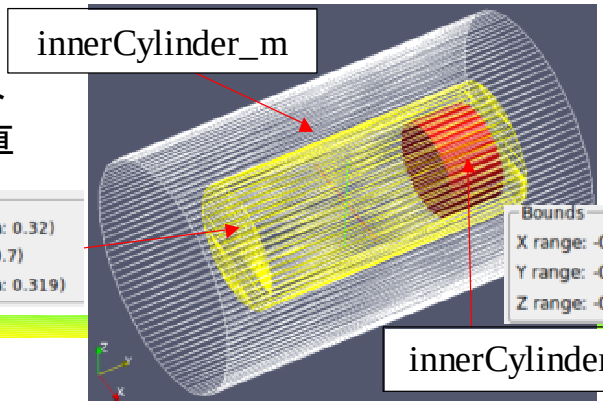
インレット側にはプロペラ軸の穴を追加



Bounds
X range: -0.3 to 0.3 (delta: 0.6)
Y range: -0.8 to 0.2 (delta: 1)
Z range: -0.3 to 0.3 (delta: 0.6)

●内部領域部分

内部領域の2つのパートはそのままstlファイルに直して利用

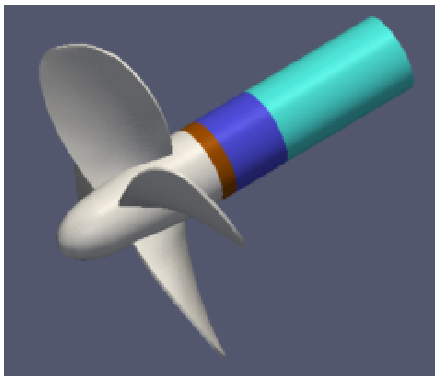


Bounds
X range: -0.16 to 0.16 (delta: 0.32)
Y range: -0.6 to 0.1 (delta: 0.7)
Z range: -0.16 to 0.16 (delta: 0.319)

Bounds
X range: -0.12 to 0.12 (delta: 0.24)
Y range: -0.08 to 0.06 (delta: 0.14)
Z range: -0.12 to 0.12 (delta: 0.24)

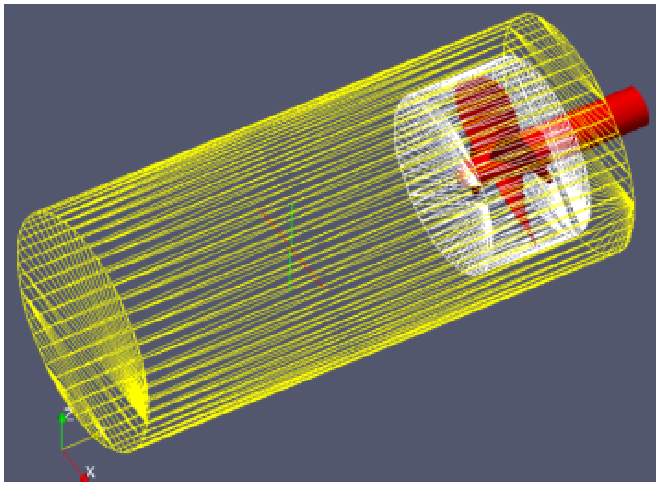
2-1．モデルの設定

●プロペラ部分



プリペラ部分もそのままstlファイルに直して利用

●内部領域とプロペラ部分



2-2. メッシュの作成

基本的なメッシュ作成は次の通り

①計算に必要なファイルの設定
②blockMesh　基礎メッシュの作成
③surfaceFeatureExtract　特徴線の取出し
④snappyHexMesh　全体メッシュの作成
⑤topoSet　ダミーのfaceSetの作成
⑥createPatch　パッチの作成
⑦createBaffles　回転面の設定
⑧mergeOrSplitBaffles　回転面の分割
⑨boundaryの調整

2-2. メッシュの作成

①計算に必要なファイルの設定

```
CASE_DIR=case_set/case1/set
BLOCK_MESH=blockMeshDict_case1
DECOMPOSE_PAR=decomposeParDict_case1_mesh
SNAPPYHEXMESH_DICT=snappyHexMeshDict_case1
CONTROL_DICT=controlDict_case1_mesh
DummyfaseSet=DummyfaceSet.topoSetDict
CREATE_PatchDict=createPatchDict
AMI_AREA=innerCylinderSmall
```

②基礎メッシュの作成（blockMeshの実施）

```
echo "blockMesh"
cp -r  $CASE_DIR/$CONTROL_DICT system/controlDict
cp -r  $CASE_DIR/$BLOCK_MESH constant/polyMesh/blockMeshDict
$runApplication blockMesh > log.blockMesh
```

③特徴線の取出し（surfaceFeatureExtractの実施）

```
echo "surfaceFeatureExtract"
$runApplication surfaceFeatureExtract -includedAngle 150 -minElem 10 constant/triSurface/innerCylinderSmall_m.stl   innerCylinderSmall_m
$runApplication surfaceFeatureExtract -includedAngle 150 -minElem 10 constant/triSurface/outerCylinder_m.stl        outerCylinder_m
$runApplication surfaceFeatureExtract -includedAngle 150 -minElem 10 constant/triSurface/propellerTip_m.stl        propellerTip_m
$runApplication surfaceFeatureExtract -includedAngle 150 -minElem 10 constant/triSurface/propellerStem2_m.stl      propellerStem2_m
$runApplication surfaceFeatureExtract -includedAngle 150 -minElem 10 constant/triSurface/propellerStem3_m.stl      propellerStem3_m
```