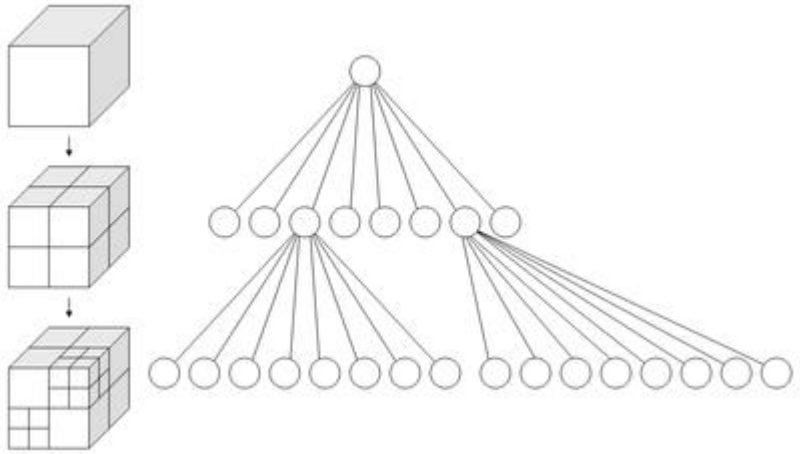
cfMeshによるメッシュ作成入門

秋山善克

cfMesh

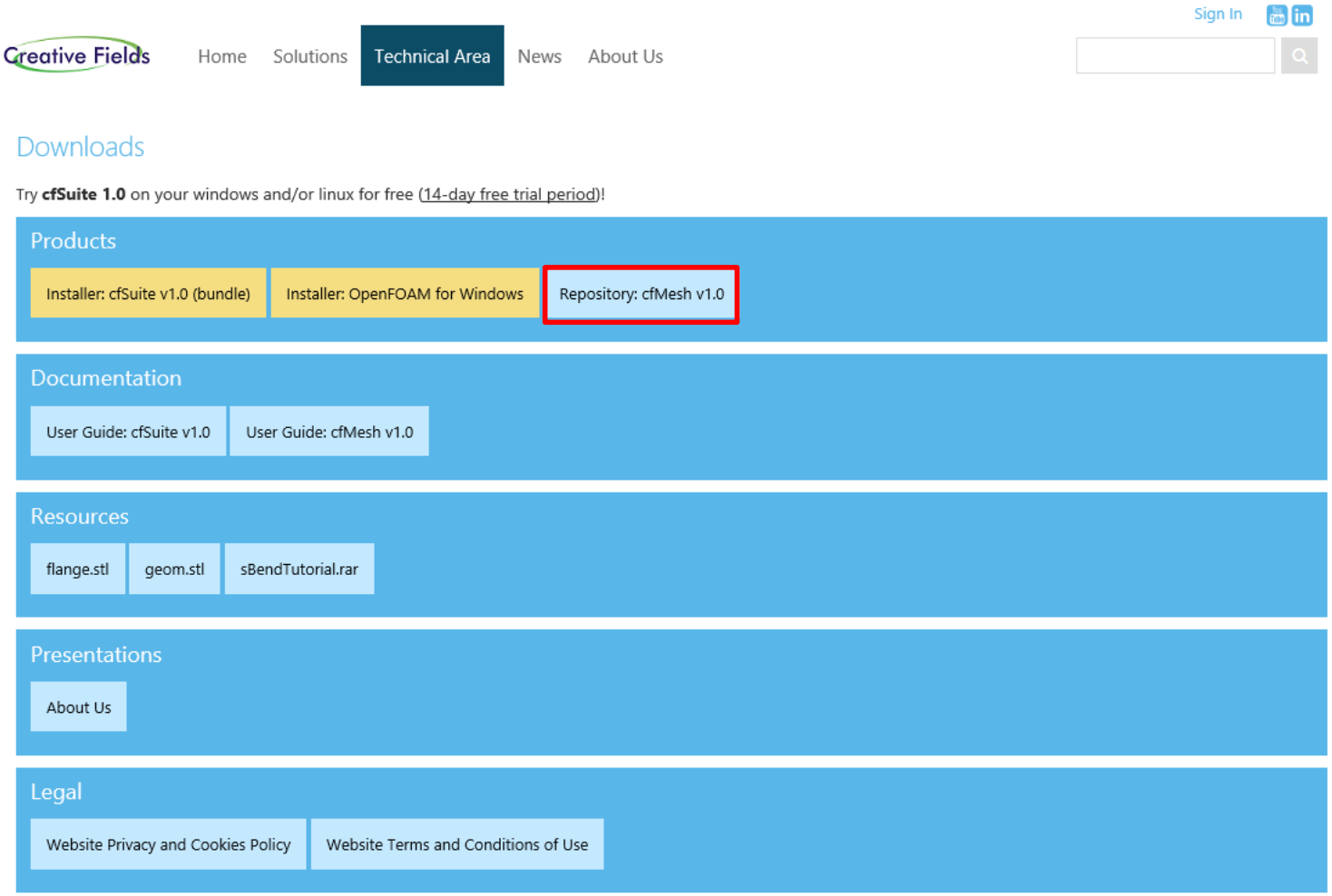
- 八分木法によるメッシュ生成
  - 最大セルサイズを基準とした細分化を実施
- STL表面や基礎形状で細分化領域を指定できる
  - 八分木法により指定領域はセルサイズを下回る大きさまで細分化される
- STL表面に適合するようにメッシュを生成
- STL表面に境界層挿入できる
  - 一括挿入も可能
  - STL領域毎に指定することも可能



http://ja.wikipedia.org/wiki/%E5%85%AB%E5%88%86%E6%9C%A8

cfMeshのダウンロード

http://www.c-fields.com/technical-area/downloads

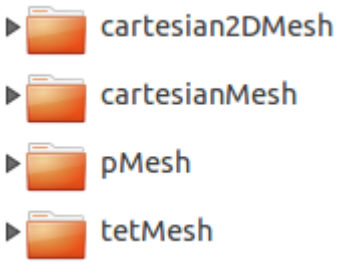


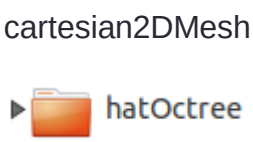
cfMeshのインストール

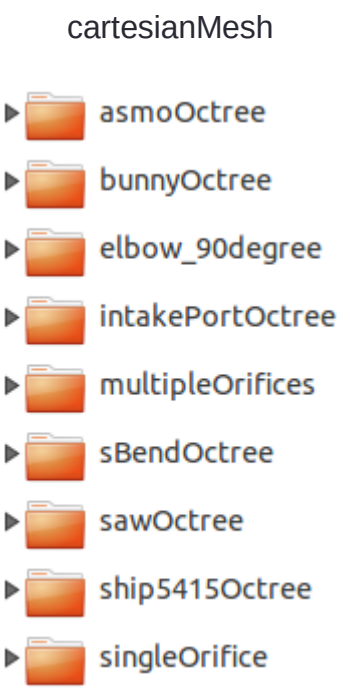
※OpenFOAM-2.3.1

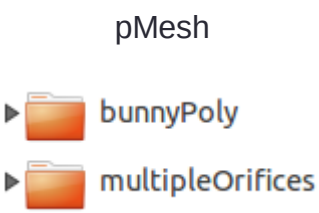
cfMesh-v1.1を展開
cfMesh-v1.1フォルダー内のAllwmakeを端末内で実行

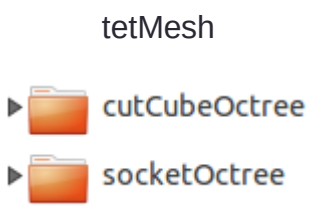
cfMeshのtutorials









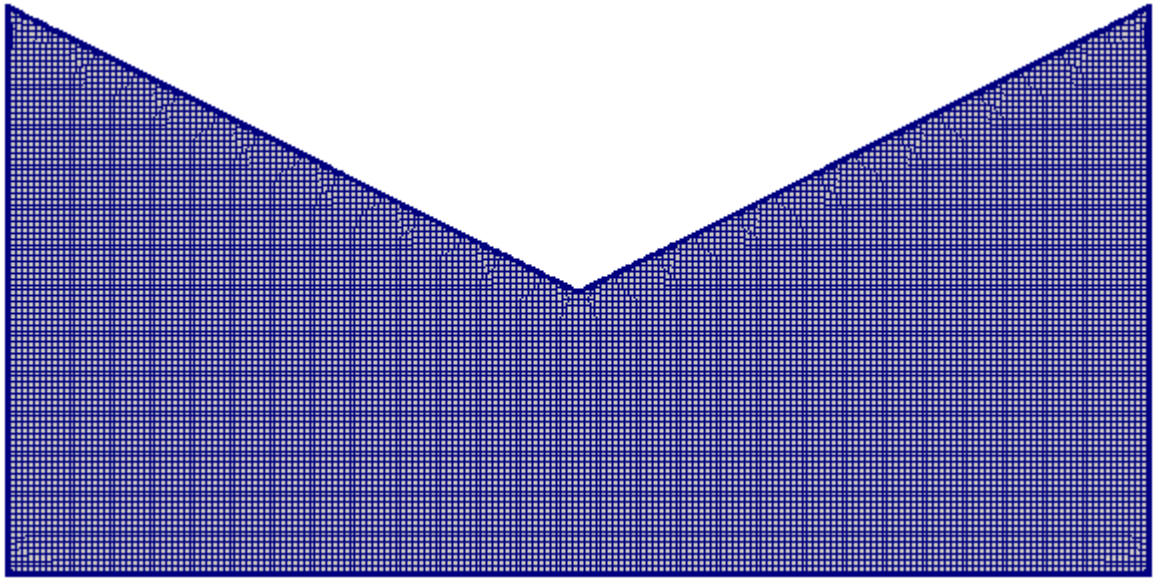


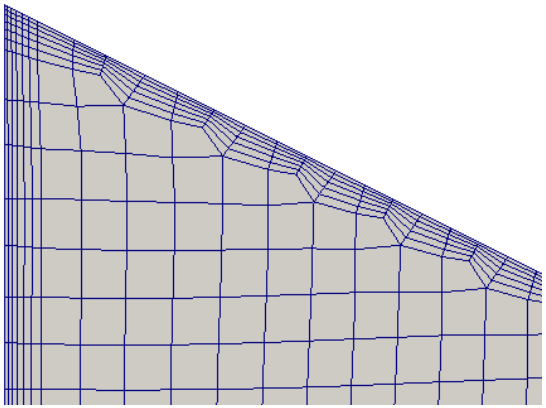
cartesian2DMesh/hatOctree

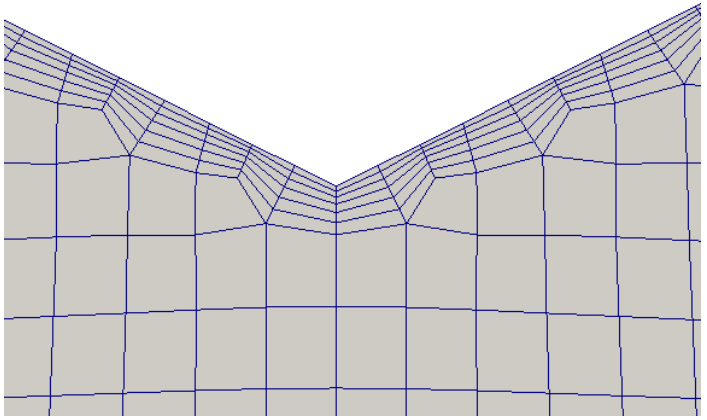
Please run cartesian2DMesh to generate a 2D mesh.

meshDict

```
maxCellSize  0.01;

surfaceFile    "geom.fms";

boundaryLayers
{
    nLayers         6;
    thicknessRatio  1.2;
}
```





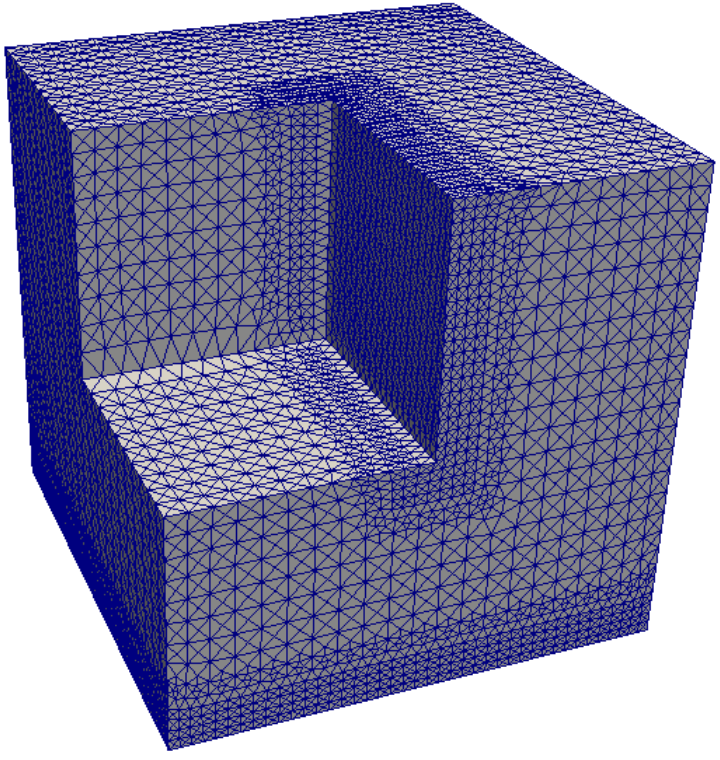


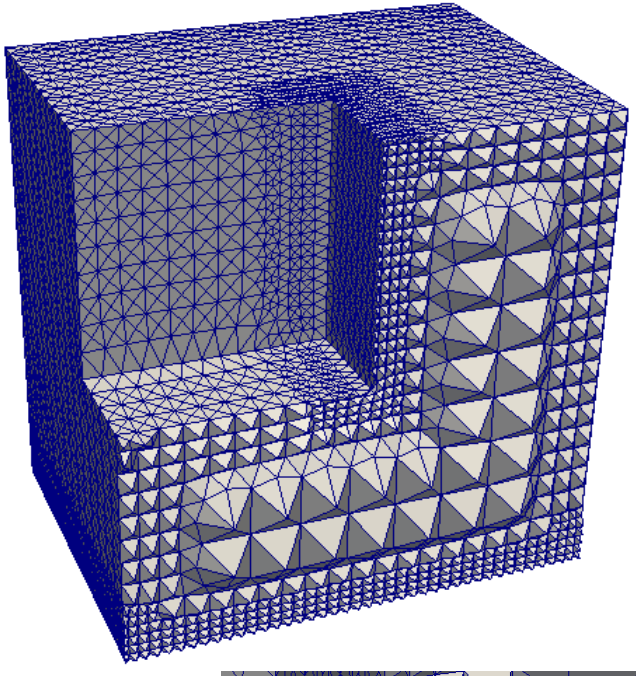
tetMesh/cutCubeOctree

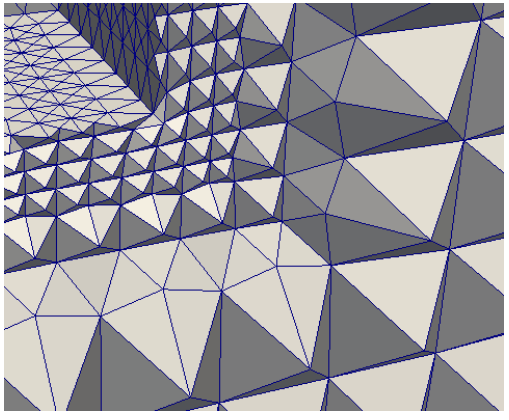
Please run tetMesh to generate a tetrahedral mesh.

meshDict

```
surfaceFile "geom1.stl";

maxCellSize 0.2;

boundaryCellSize 0.1;

minCellSize 0.1;

localRefinement
{
    patch0000
    {
        cellSize  0.05;
    }
    patch0007
    {
        cellSize  0.05;
    }
}
```







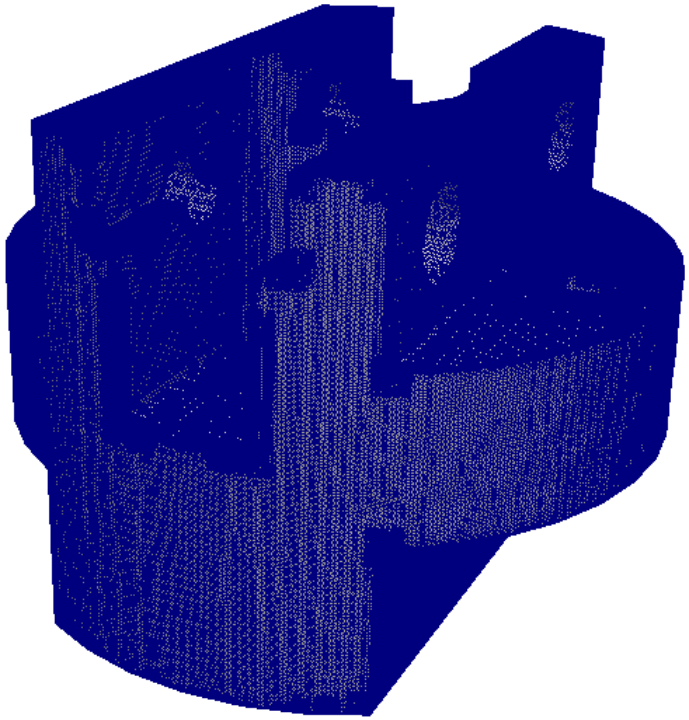
tetMesh/socketOctree

Please run cartesianMesh or tetMesh to generate te volume mesh.

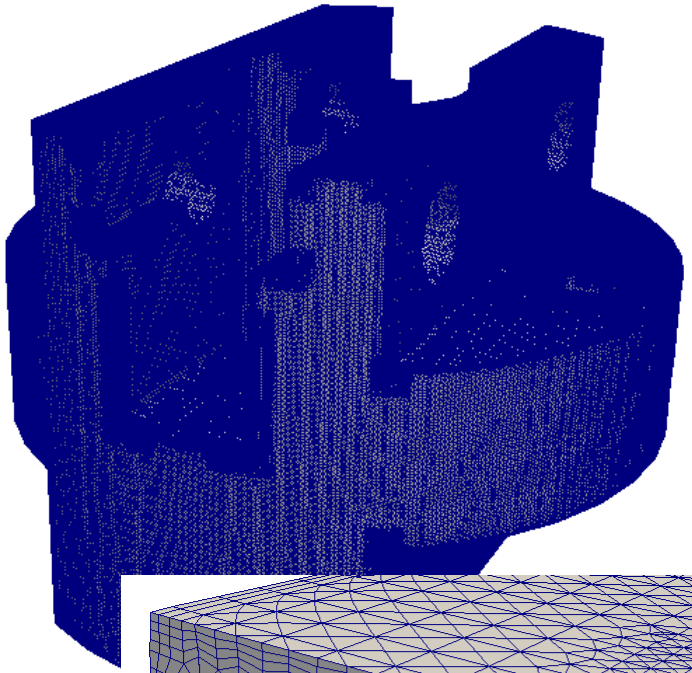
meshDict

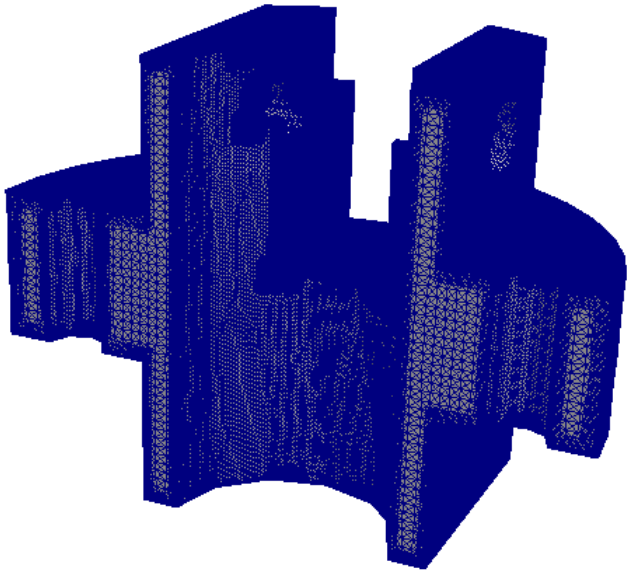
```
boundaryCellSize              1.5;
keepCellsIntersectingBoundary         1;
maxCellSize  3;
minCellSize   0.375;
removeGluedMesh               0;
surfaceFile    "socket.fms";

boundaryLayers
{
    maxFirstLayerThickness0.5;
    nLayers     3;
    thicknessRatio            1.2;

    patchBoundaryLayers
    {

        patch7
        {
            allowDiscontinuity   0;
            maxFirstLayerThickness        0.5;
            nLayers               4;
            thicknessRatio        1.1;
        }
    }
}
```

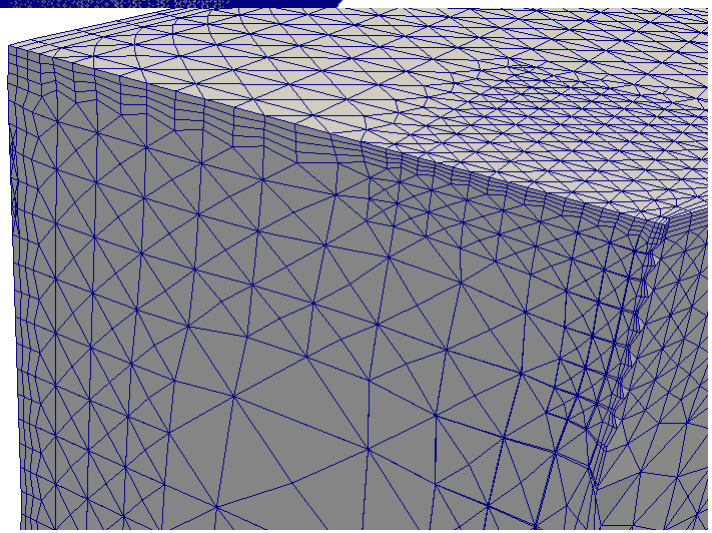
```
localRefinement
{

    patch15
    {
        additionalRefinementLevels        1;
    }

    subset1
    {
        cellSize 1.5;
    }

    subset2
    {
        cellSize 1.5;
    }
}
```

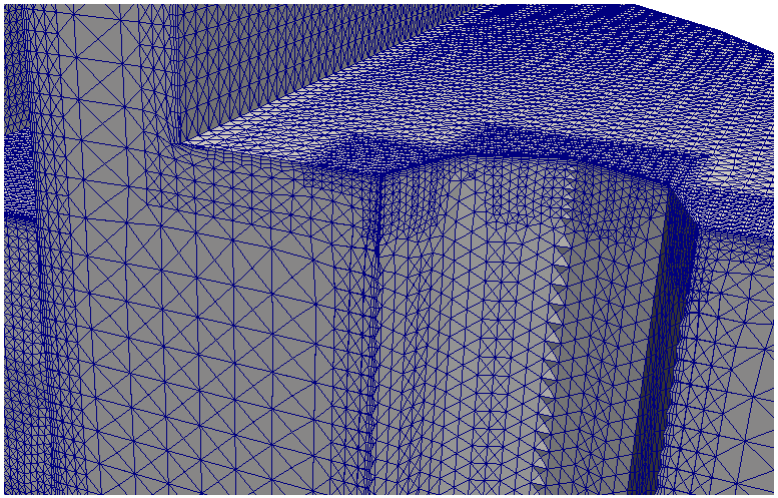


tetMesh/socketOctree







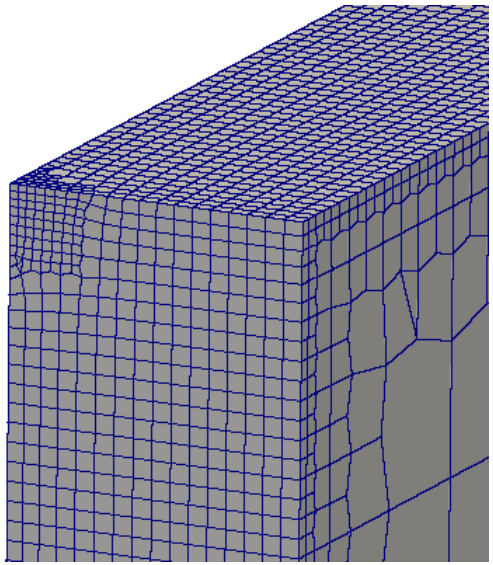


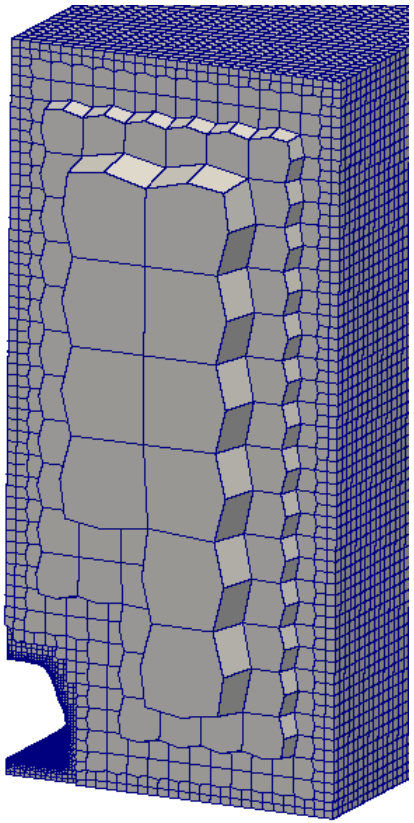
cartesianMesh/asmoOctree

Please run cartesianMesh to generate the mesh

meshDict

```
surfaceFile "geom.stl";

maxCellSize 0.2;

boundaryCellSize 0.025;

minCellSize 0.0125;

localRefinement
{
    defaultFaces0006
    {
        cellSize 0.005;
    }
    defaultFaces0007
    {
        cellSize 0.0025;
    }
    defaultFaces0009
    {
        cellSize 0.0025;
    }
}
```



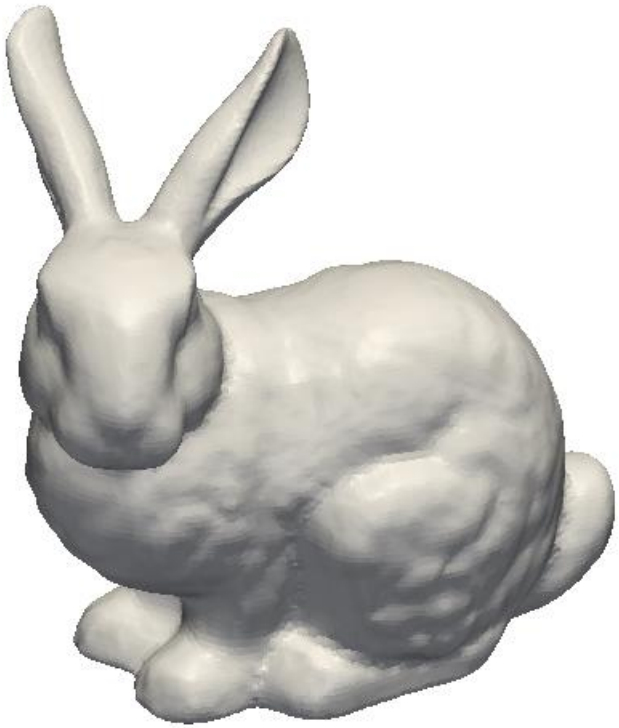


cartesianMesh/bunnyOctree

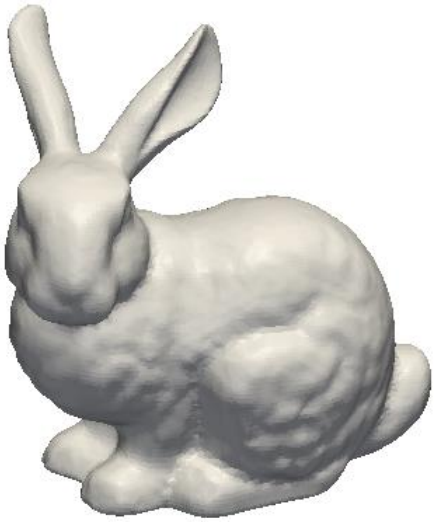
This tutorials demonstrates the usage of primitive refinement sources

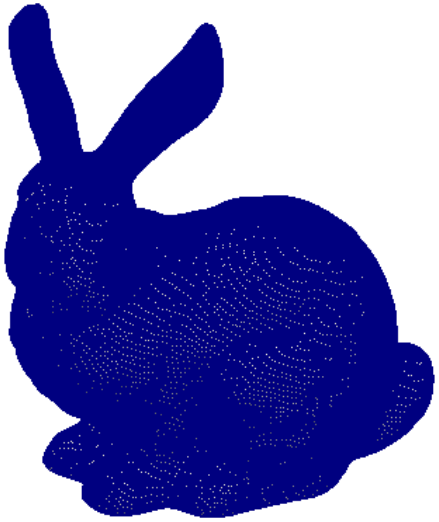
meshDict

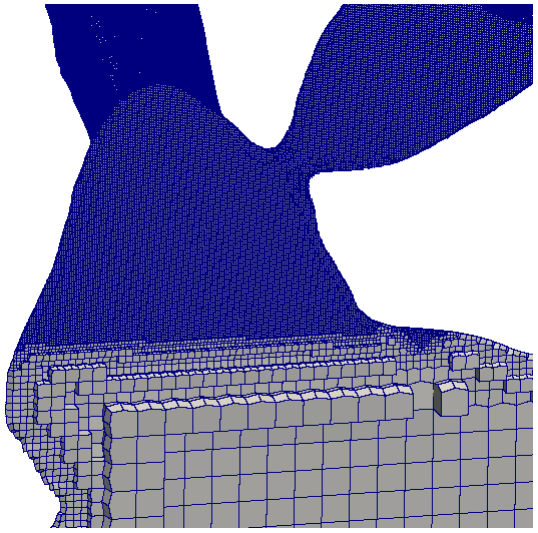
```
surfaceFile "bunnyWrapped.stl";

maxCellSize 30.0;

boundaryCellSize 7.5;

objectRefinements
{
    ear1
    {
        cellSize 3.75;
        type cone;
        p0   (-100 1873 -320);
        radius0  200;
        p1   (-560 1400 0);
        radius1   200;
    }
    ear2
    {
        cellSize 3.75;
        type cone;
        p0   (-650 1873 -620);
        radius0  200;
        p1   (-670 1300 0);
        radius1  200;
    }
    tail
    {
        cellSize 3.75;
        type box;
        centre  (500 500 150);
        lengthX  100;
        lengthY  150;
        lengthZ  200;
    }
    insideTheBody
    {
        cellSize 3.75;
        type sphere;
        centre  (0 700 0);
        radius  50;
    }
    muzzlePiercing
    {
        cellSize 3.75;
        type line;
        p0      (-750 1000 450);
        p1      (-750 1500 450);
    }
}
```

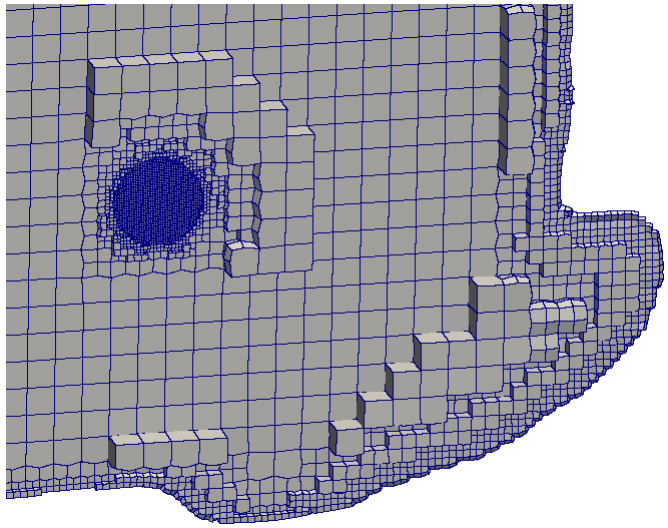


cartesianMesh/bunnyOctree









cartesianMesh/elbow_90degree

追加

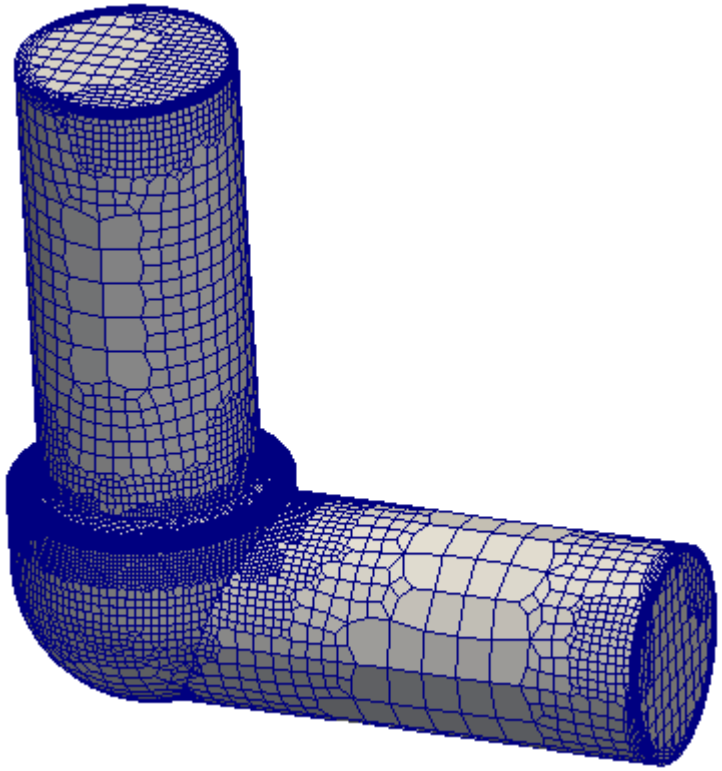
readme

cfMesh Example Case
Date: 02 October 2014
Application: cartesianMesh

Goal: Demonstration of the regular expressions feature available within
cfMesh for specifying patch names in the meshDict file.

STL File: elbow_90degree.stl
STL Type: Multi-solid

Patches within the STL File (Note: Each patch is one STL Solid):
inlet_S73
outlet_S74
bendOuter_S75
bendOuter_S76
bendInner_S77
ringArea_S78
fixedWalls_S79
fixedWalls_S80
fixedWalls_S81



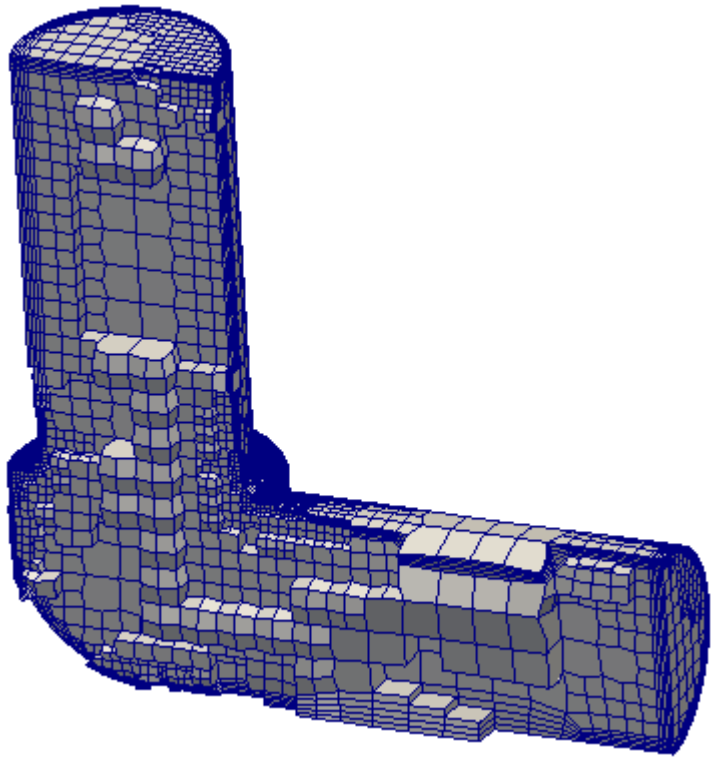
cartesianMesh/elbow_90degree

追加

meshDict

```
surfaceFile "elbow_90degree.stl";

maxCellSize 5.0;

boundaryCellSize 3.0;

minCellSize 1.00;

localRefinement
{
    "ringArea.*"
    {
        cellSize 0.2;
    }
}

boundaryLayers
{
    nLayers 5;

    thicknessRatio 1.1;

    maxFirstLayerThickness 0.5;

//    patchBoundaryLayers
//    {
//    }
}
```

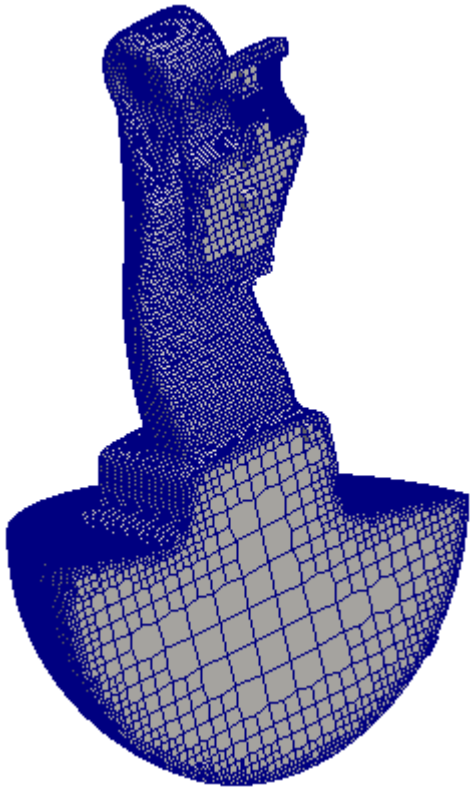
```
renameBoundary
{
    defaultName     fixedWalls;
    defaultType     wall;

    newPatchNames
    {
        "inlet.*"
        {
            newName     inlet;
            newType     patch;
        }

        "outlet.*"
        {
            newName     outlet;
            newType     patch;
        }
    }
}
```



cartesianMesh/intakePortOctree

meshDict

```
surfaceFile "geom2.stl";

maxCellSize 10;

boundaryCellSize 1.25;

minCellSize 2.0;

localRefinement
{
    patch001
    {
        cellSize 0.625;
    }
    patch002
    {
        cellSize 0.625;
    }
    patch003
    {
        cellSize 0.625;
    }
    patch004
    {
        cellSize 0.625;
    }
    patch005
    {
        cellSize 0.625;
    }
    patch006
    {
        cellSize 0.625;
    }
    patch007
    {
        cellSize 0.625;
    }
    patch008
    {
        cellSize 0.625;
    }
}
```



cartesianMesh/multipleOrifices

追加

readme

cfMesh Example Case
Date: 02 October 2014
Application: cartesianMesh

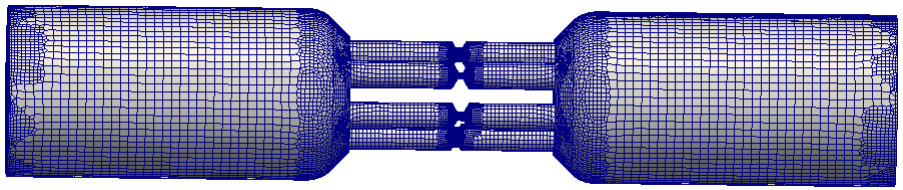
Goal: Demonstration of the regular expressions feature available within
cfMesh for specifying patch names in the meshDict file.

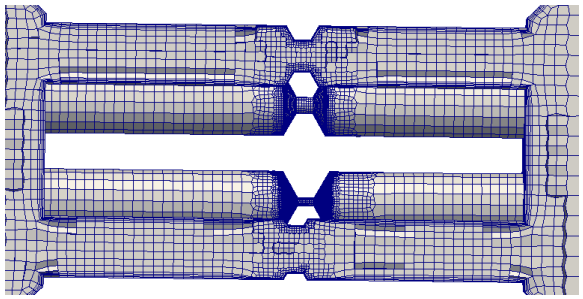
STL File: multipleOrifices.stl
STL Type: Multi-solid

Patches within the STL File (Note: Each patch is one STL Solid):
inlet_S42
outlet_S43

orifice01_S44
orifice01_S45
orifice01_S46
orifice02_S47
orifice02_S48
orifice02_S49
orifice03_S50
orifice03_S51
orifice03_S52
orifice04_S53
orifice04_S54
orifice04_S55
orifice05_S56
orifice05_S57
orifice05_S58
orifice06_S59
orifice06_S60
orifice06_S61

tubes_S62
tubes_S63
tubes_S64
tubes_S65
tubes_S66
tubes_S67
tubes_S68
tubes_S69
tubes_S70
tubes_S71
tubes_S72
tubes_S73
fixedWalls_S74
fixedWalls_S75
fixedWalls_S76
fixedWalls_S77
fixedWalls_S78
fixedWalls_S79
fixedWalls_S80
fixedWalls_S81





cartesianMesh/multipleOrifices

追加

```
surfaceFile "multipleOrifices.stl";
maxCellSize 5.0;
boundaryCellSize 2.0;
minCellSize 1.00;
localRefinement
{
    "orifice01.*"
    {
        cellSize 0.1;
    }

    "orifice02.*"
    {
        cellSize 0.2;
    }

    "orifice0[3-6].*"
    {
        cellSize 0.3;
    }
}
```

```
boundaryLayers
{
//    nLayers 3;
//    thicknessRatio 1.2;
//    maxFirstLayerThickness 0.5;
    patchBoundaryLayers
    {
        "orifice.*"
        {
            nLayers           4;
            thicknessRatio    1.2;
            maxFirstLayerThickness 0.2;
            allowDiscontinuity 0;
        }
        "fixedWalls.*"
        {
            nLayers           4;
            thicknessRatio    1.2;
            maxFirstLayerThickness 1.0;
            allowDiscontinuity 0;
        }
        "tubes.*"
        {
            nLayers           4;
            thicknessRatio    1.2;
            maxFirstLayerThickness 1.0;
            allowDiscontinuity 0;
        }
    }
 optimiseLayer 1;
}
```

```
renameBoundary
{
    defaultName     fixedWalls;
    defaultType     wall;

    newPatchNames
    {
        "inlet.*"
        {
            newName     inlet;
            newType     patch;
        }

        "outlet.*"
        {
            newName     outlet;
            newType     patch;
        }
    }
}
```

cartesianMesh/sBendOctree

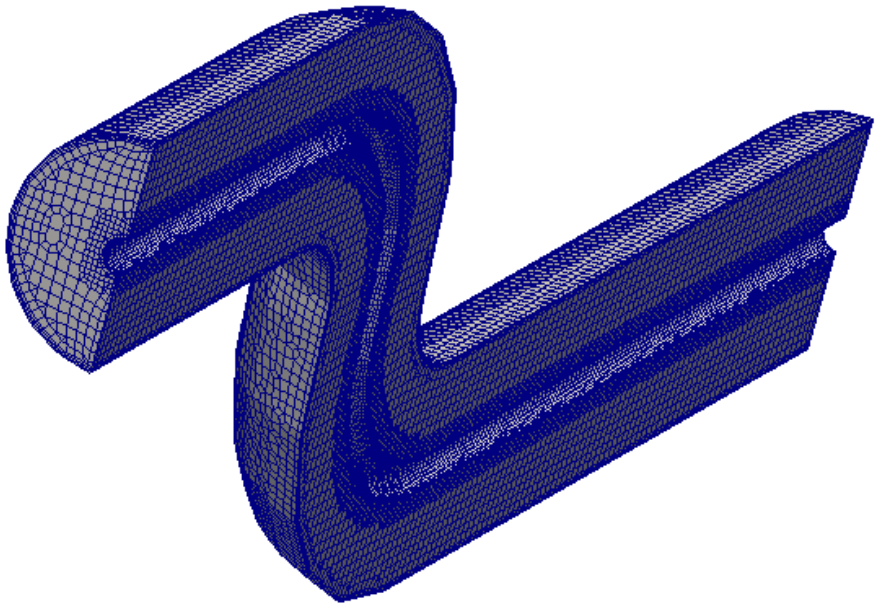
修正

The exmaple demonstrates usage of subsets for refinement, and how to set up boundary layer properties. To generate the mesh please run cartesianMesh or tetMesh.

```
maxCellSize  0.1;
surfaceFile    "sBend.fms";
boundaryLayers
{
    nLayers     1;
    patchBoundaryLayers
    {
        walls
        {
            nLayers             3;
            thicknessRatio      1.2;
        }
    }

    optimiseLayer 1;

    optimisationParameters
    {
        nSmoothNormals 5;
        relThicknessTol 0.15;
        featureSizeFactor 0.3;
        reCalculateNormals 1;
        maxNumIterations 5;
    }
}
```

optimisationParametersが追加されている

```
localRefinement
{

    refFine
    {
        cellSize 0.025;
    }

    walls
    {
        cellSize 0.05;
    }
}
```



Allrun

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

runApplication cartesianMesh
runApplication improveSymmetryPlanes
runApplication checkMesh
```

Allrunが動くように修正されている
improveSymmetryPlanesが追加

cartesianMesh/sawOctree

修正

Please run cartesianMesh to generate the mesh.

meshDict

```
surfaceFile "sav1.stl";

maxCellSize 0.25;

boundaryCellSize 0.125;
```

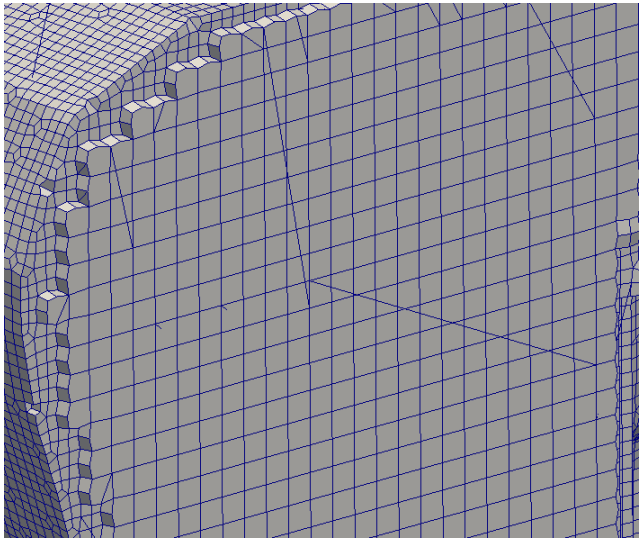
decomposeParDict

```
numberOfSubdomains  4;
```

Allrun

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

runApplication preparePar
runParallel `which cartesianMesh` 4
runParallel `which checkMesh` 4
if [ "$WM_PROJECT" = "OpenFOAM" ]
then
runApplication reconstructParMesh -constant -fullMatch
else
runApplication reconstructParMesh -zeroTime
fi
```

Allrunが並列で動くように修正されている



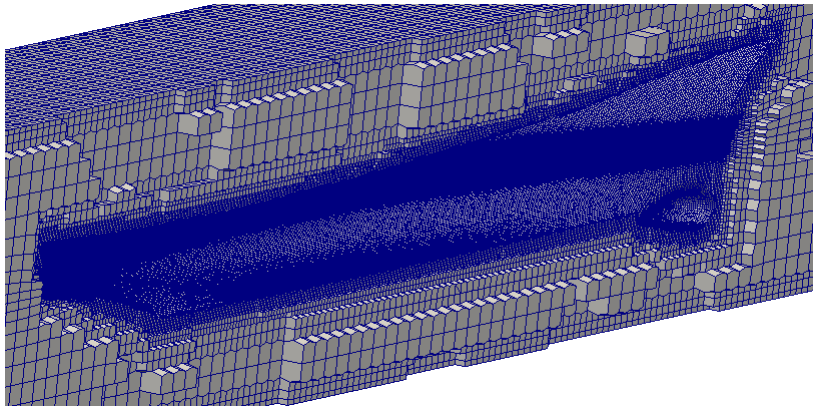
cartesianMesh/ship5415Octree

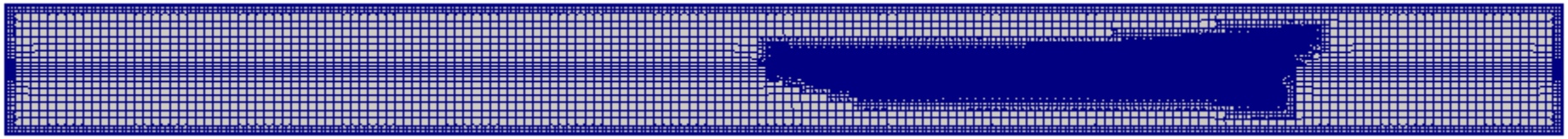
追加

This tutorials demonstrates the usage of surface mesh refinement sources

Allrun

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

runApplication surfaceFeatureEdges 5415Joined2.stl 5415Joined3.ftr

runApplication cartesianMesh
runApplication checkMesh
```





cartesianMesh/ship5415Octree

追加

meshDict

```
surfaceFile "5415Joined3.ftr";
maxCellSize 100;
boundaryCellSize    50;
//boundaryCellSizeRefinementThickness 50;

surfaceMeshRefinement
{
    hull
    {
        additionalRefinementLevels  3;
        surfaceFile "5415Joined1.stl";
        refinementThickness 50;
    }
}

anisotropicSources
{
    Box
    {
        type box;
        centre (2800 0 250);
        lengthX 6000;
        lengthY 1000;
        lengthZ 200;
        scaleX 1;
        scaleY 1;
        scaleZ 0.3;
    }
```

```
/*
    planeUpper
    {
            type plane;
            normal (0 0 1);
            origin (0 0 250);
            scalingDistance 125;
            scalingFactor 0.5;
    }
    planeLower
    {
            type plane;
            normal (0 0 -1);
            origin (0 0 250);
            scalingDistance 125;
            scalingFactor 0.5;
    }
*/
}
```

```
boundaryLayers
{
    patchBoundaryLayers
    {
        HULL_AND_BOX_1
        {
            nLayers 5;
            thicknessRatio 1.1;
        }
    }

    optimiseLayer 1;

    optimisationParameters
    {
        nSmoothNormals 3;
        maxNumIterations 5;
        featureSizeFactor 0.4;
        reCalculateNormals 1;
        relThicknessTol 0.1;
    }
}
```

cartesianMesh/singleOrifice

追加

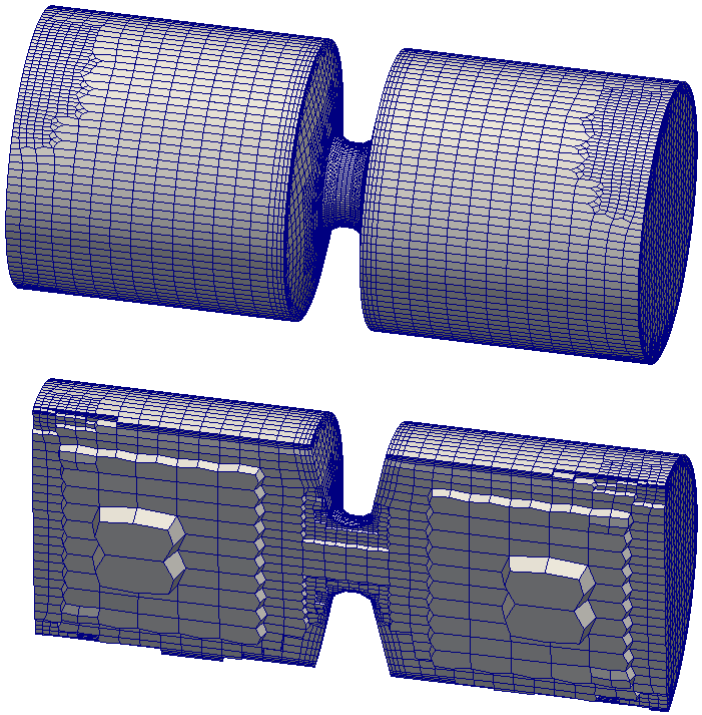
readme

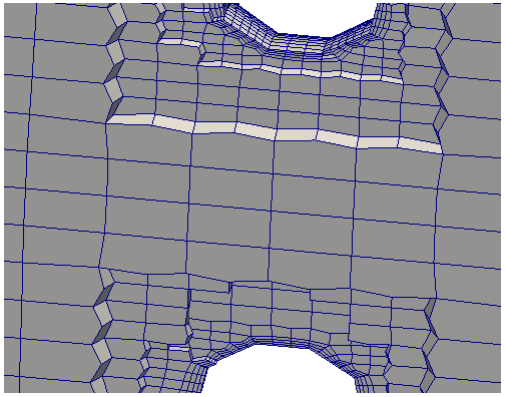
cfMesh Example Case
Date: 02 October 2014
Application: cartesianMesh

Goal: Demonstration of the regular expressions feature available within
cfMesh for specifying patch names in the meshDict file.

STL File: singleOrifice.stl
STL Type: Multi-solid

Patches within the STL File (Note: Each patch is one STL Solid):
inlet_S11
outlet_S12
orificeRegion_S13
orificeRegion_S14
orificeRegion_S15
fixedWalls_S16
fixedWalls_S17
fixedWalls_S18
fixedWalls_S19





cartesianMesh/singleOrifice

追加

meshDict

```
surfaceFile "singleOrifice.stl";

maxCellSize 3.0;

boundaryCellSize 1.0;

minCellSize 0.50;

anisotropicSources
{
    Plane
    {
        type plane;
        origin (0 0 -20);
        normal (0 0 1);
        scalingDistance 45;
        scalingFactor 2;
    }
}

localRefinement
{
    "orificeRegion.*"
    {
        cellSize 0.2;
    }
}
```

```
boundaryLayers
{
//    nLayers 3;
//    thicknessRatio 1.2;
//    maxFirstLayerThickness 0.5;

    patchBoundaryLayers
    {
        "orificeRegion.*"
        {
            nLayers           4;
            thicknessRatio    1.2;
            maxFirstLayerThickness 0.2;
            allowDiscontinuity 0;
        }

                "fixedWalls.*"
        {
            nLayers           4;
            thicknessRatio    1.2;
            maxFirstLayerThickness 0.5;
            allowDiscontinuity 0;
        }
    }
}
```

```
renameBoundary
{
    defaultName         fixedWalls;
    defaultType         wall;

    newPatchNames
    {
        "orificeRegion.*"
        {
            newName    orificeRegion;
            newType    wall;
        }

        "inlet.*"
        {
            newName    inlet;
            newType    patch;
        }

        "outlet.*"
        {
            newName    outlet;
            newType    patch;
        }
    }
}
```

pMesh/bunnyPoly

追加

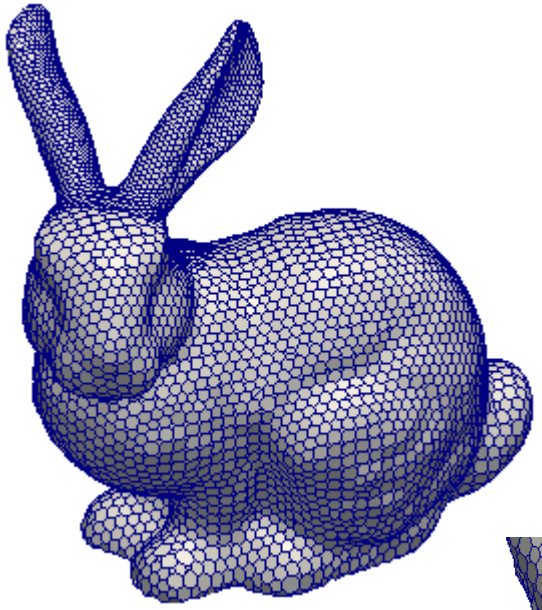
This tutorial demonstrates the polyhedral meshing workfow without boundary layers. It also demonstrates usage of box, line, sphere and cone refinement sources.

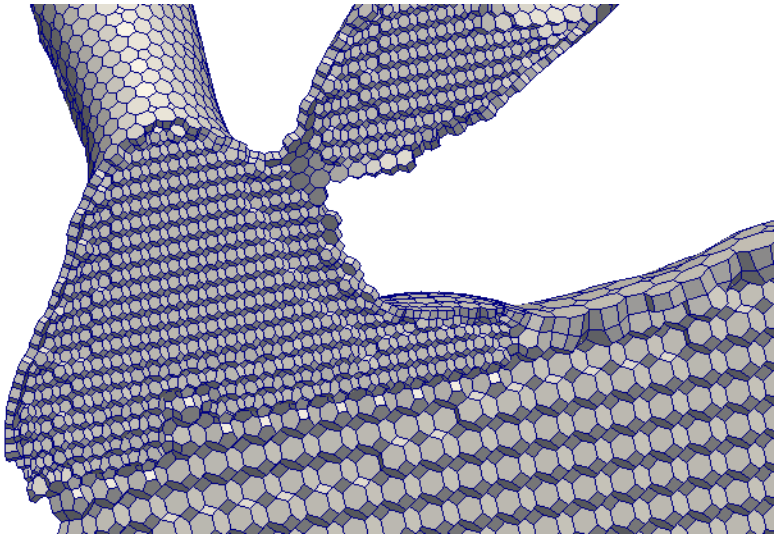
meshDict

```
maxCellSize  40;
surfaceFile  "bunnyWrapped.stl";

objectRefinements
{

ear1
{
cellSize     20.1;
p0           ( -100 1873 -320 );
p1           ( -560 1400 0 );
radius0      200;
radius1      200;
type         cone;
}

ear2
{
cellSize     20.1;
p0           ( -650 1873 -620 );
p1           ( -670 1300 0 );
radius0      200;
radius1      200;
type         cone;
}

/*
insideTheBody
{
cellSize     20.1;
centre       ( 0 700 0 );
radius       50;
refinementThickness     50;
type         sphere;
}

muzzlePiercing
{
cellSize     20.1;
p0           ( -750 1000
450 );
p1           ( -750 1500
450 );
type         line;
}

tail
{
cellSize     20.1;
centre       ( 500 500 150 );
lengthX      100;
lengthY      150;
lengthZ      200;
type         box;
}
*/
}

workflowControls
{
}
```

pMesh/bunnyPoly

追加





pMesh/multipleOrifices

追加

readme

cfMesh Example Case
Date: 08 June 2015
Application: pMesh

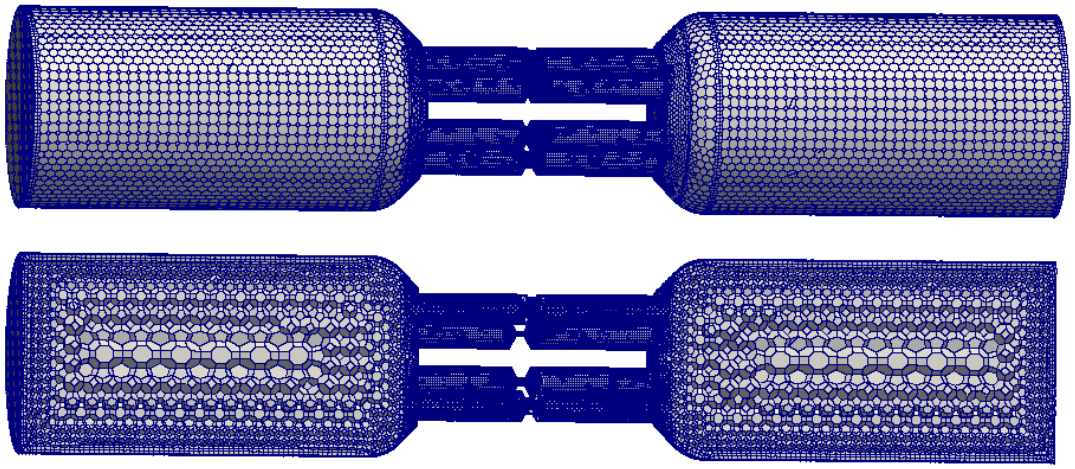
Goal: Demonstration of the regular expressions feature available within cfMesh for specifying patch names in the meshDict file.

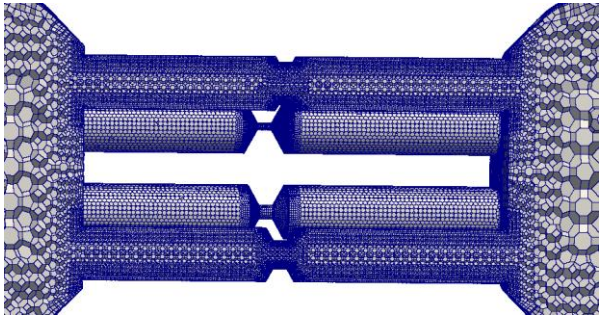
STL File: multipleOrifices.stl
STL Type: Multi-solid

Patches within the STL File (Note: Each patch is one STL Solid):
inlet_S42
outlet_S43

orifice01_S44
orifice01_S45
orifice01_S46
orifice02_S47
orifice02_S48
orifice02_S49
orifice03_S50
orifice03_S51
orifice03_S52
orifice04_S53
orifice04_S54
orifice04_S55
orifice05_S56
orifice05_S57
orifice05_S58
orifice06_S59
orifice06_S60
orifice06_S61

tubes_S62
tubes_S63
tubes_S64
tubes_S65
tubes_S66
tubes_S67
tubes_S68
tubes_S69
tubes_S70
tubes_S71
tubes_S72
tubes_S73
fixedWalls_S74
fixedWalls_S75
fixedWalls_S76
fixedWalls_S77
fixedWalls_S78
fixedWalls_S79
fixedWalls_S80
fixedWalls_S81





pMesh/multipleOrifices

追加

meshDict

```
surfaceFile "multipleOrifices.stl";

maxCellSize 5.0;

boundaryCellSize 2.0;

localRefinement
{
    "orifice01.*"
    {
        cellSize 0.1;
    }

    "orifice02.*"
    {
        cellSize 0.2;
    }

    "orifice0[3-6].*"
    {
        cellSize 0.3;
    }

    "tubes.*"
    {
        cellSize 0.4;
    }
}
```

```
boundaryLayers
{
    nLayers 1;
    thicknessRatio 1.2;
    patchBoundaryLayers
    {
        "orifice.*"
        {
            nLayers          4;
            thicknessRatio   1.2;
            axFirstLayerThickness 0.2;
            allowDiscontinuity 0;
        }
        "fixedWalls.*"
        {
            nLayers          4;
            thicknessRatio   1.2;
            maxFirstLayerThickness 1.0;
            allowDiscontinuity 0;
        }
        "tubes.*"
        {
            nLayers          4;
            thicknessRatio   1.2;
            maxFirstLayerThickness 1.0;
            allowDiscontinuity 0;
        }
    }
    optimiseLayer 1;
}
```

```
renameBoundary
{
    defaultName     fixedWalls;
    defaultType     wall;

    newPatchNames
    {
        "inlet.*"
        {
            newName     inlet;
            newType     patch;
        }
        "outlet.*"
        {
            newName     outlet;
            newType     patch;
        }
    }
}
```

```
workflowControls
{
//stopAfter templateGeneration;
//stopAfter surfaceTopology;
//stopAfter surfaceProjection;
//stopAfter patchAssignment;
//stopAfter edgeExtraction;
//stopAfter boundaryLayerGeneration;
//stopAfter meshOptimisation;
//stopAfter boundaryLayerRefinement;

//restartFromLatestStep 1;
}
```

本日の演習内容

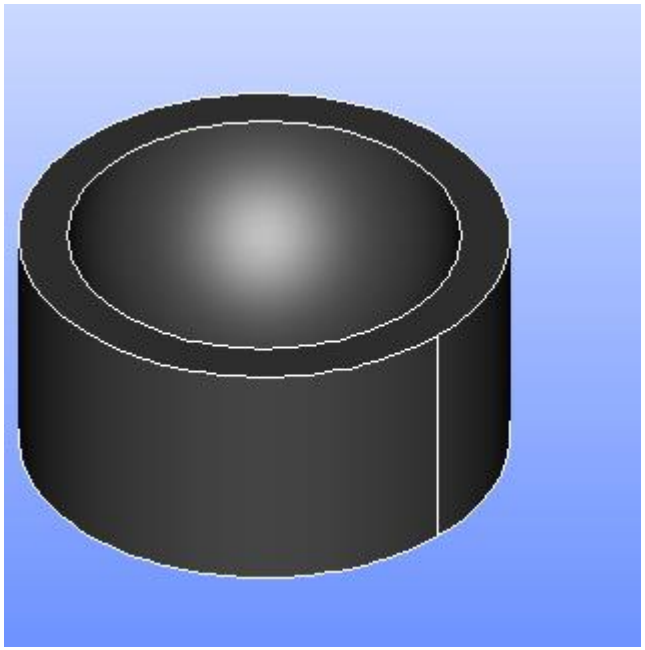
- 演習1　GeometryからcfMesh作成
- 演習2　表面メッシュを作成したのちcfMesh作成
- 演習3　境界層の作成（ポリヘドラルメッシュの作成）
- 演習4　部分的なセルサイズの指定
- 演習5　異方性メッシュの作成
- 演習6　欠けた形状のメッシュ作成
- 演習7　snappyHexMeshとの比較

演習1　Primitivesによるモデル作成

①XY平面を底面基準とし、Z軸を中心軸とする半径50mm、高さ50mmの円柱を作成する。（ソリッドモデルA）

②座標値(0,0,50)を中心とする半径40mmの球形状を作成する。（ソリッドモデルB）

③円柱（ソリッドモデルA）と球（ソリッドモデルB）を組み合わせる。

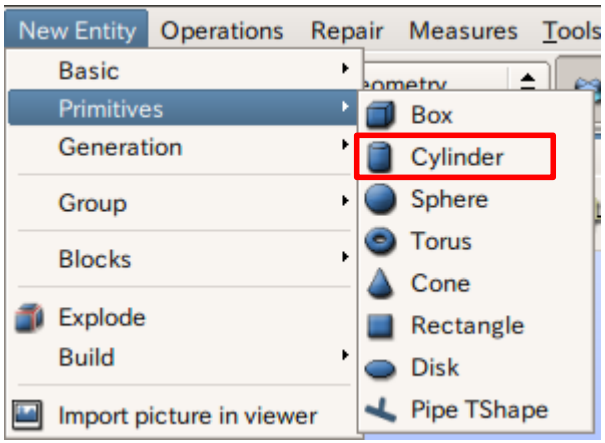


演習1　Primitivesによるモデル作成

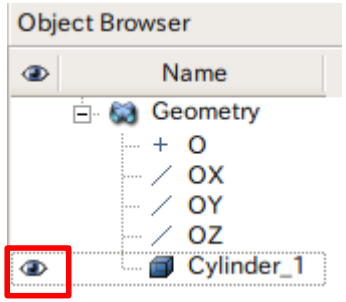
①XY平面を底面基準とし、Z軸を中心軸とする半径50mm、高さ50mmの円柱を作成する。（ソリッドモデルA）

円柱の作成

New Entity>Primitives>Cylinder



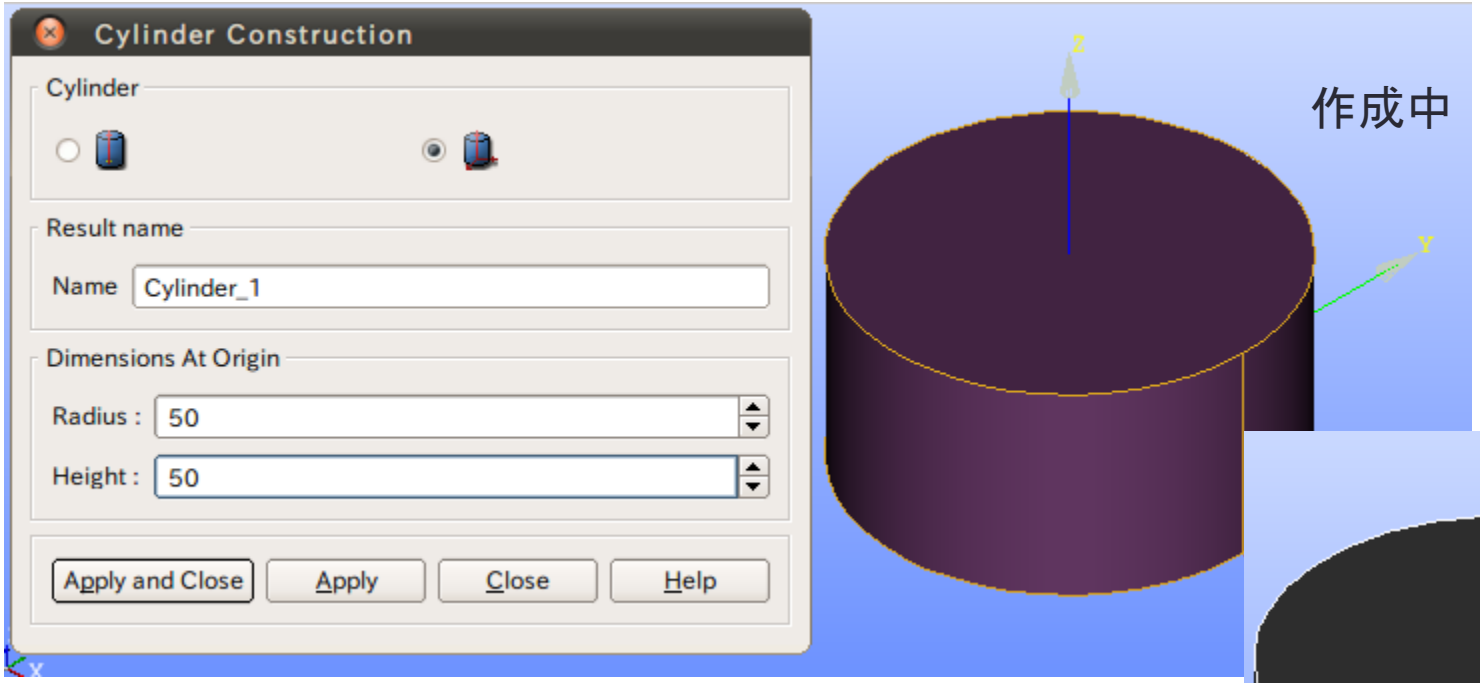
連続して作成する場合はApply

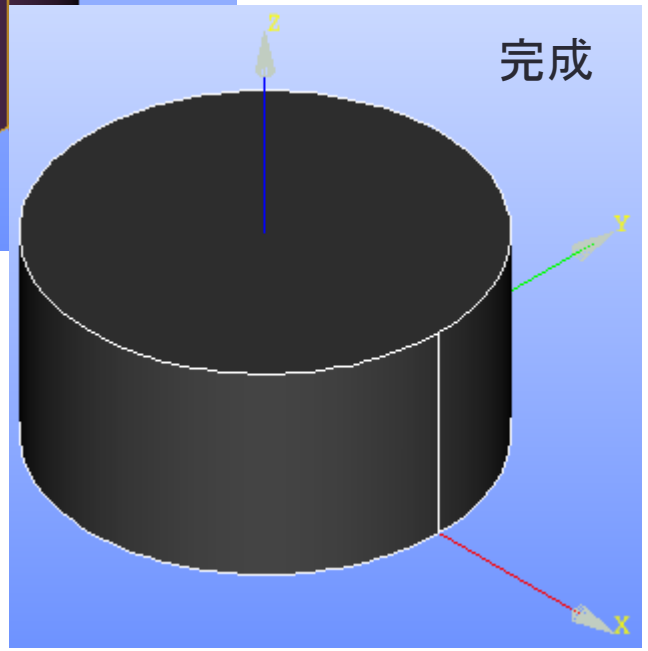


オブジェクトブラウザに追加される

表示/非表示切り替え

演習1　Primitivesによるモデル作成



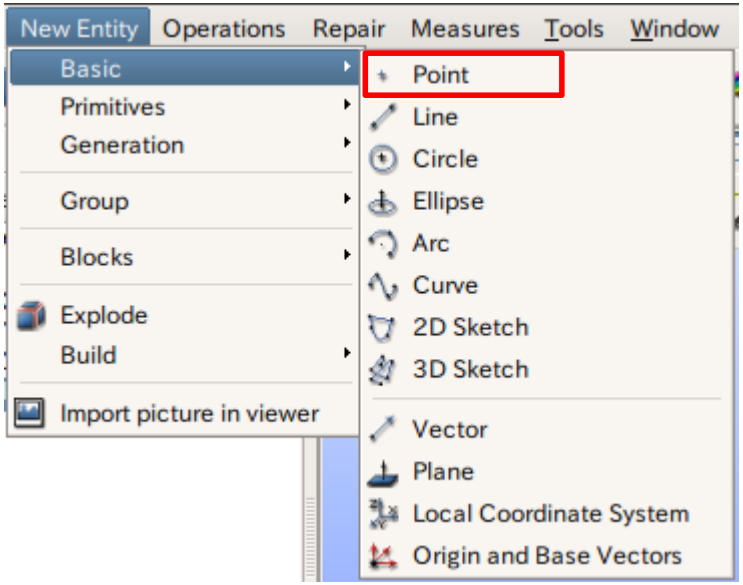


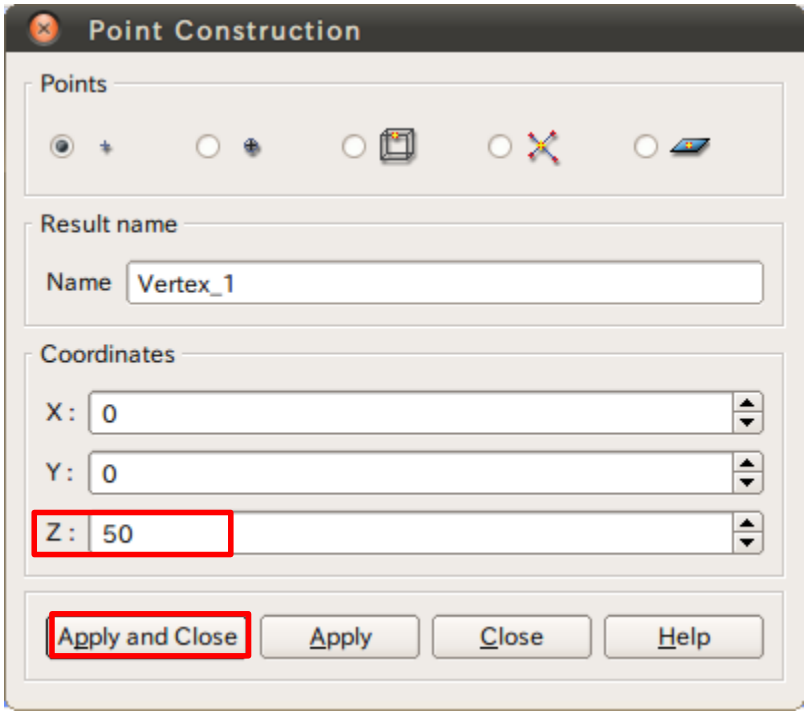
演習1　Primitivesによるモデル作成

②座標値(0,0,50)を中心とする半径40mmの球形状を作成する。(ソリッドモデルB)

点の作成

New Entity>Basic>Point



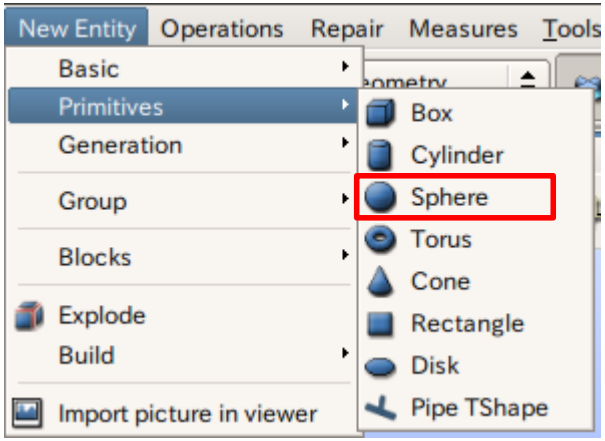


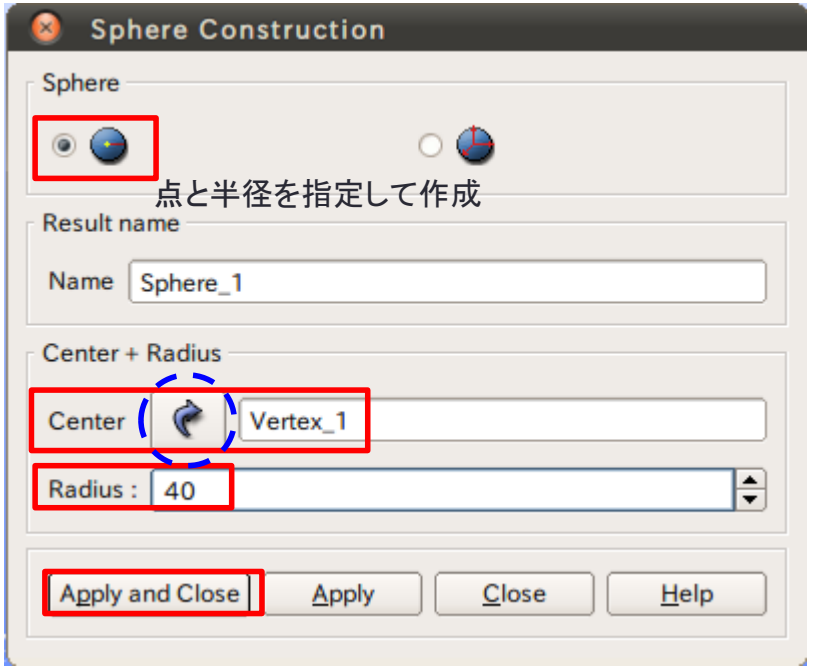
演習1　Primitivesによるモデル作成

②座標値(0,0,50)を中心とする半径40mmの球形状を作成する。(ソリッドモデルB)

球の作成

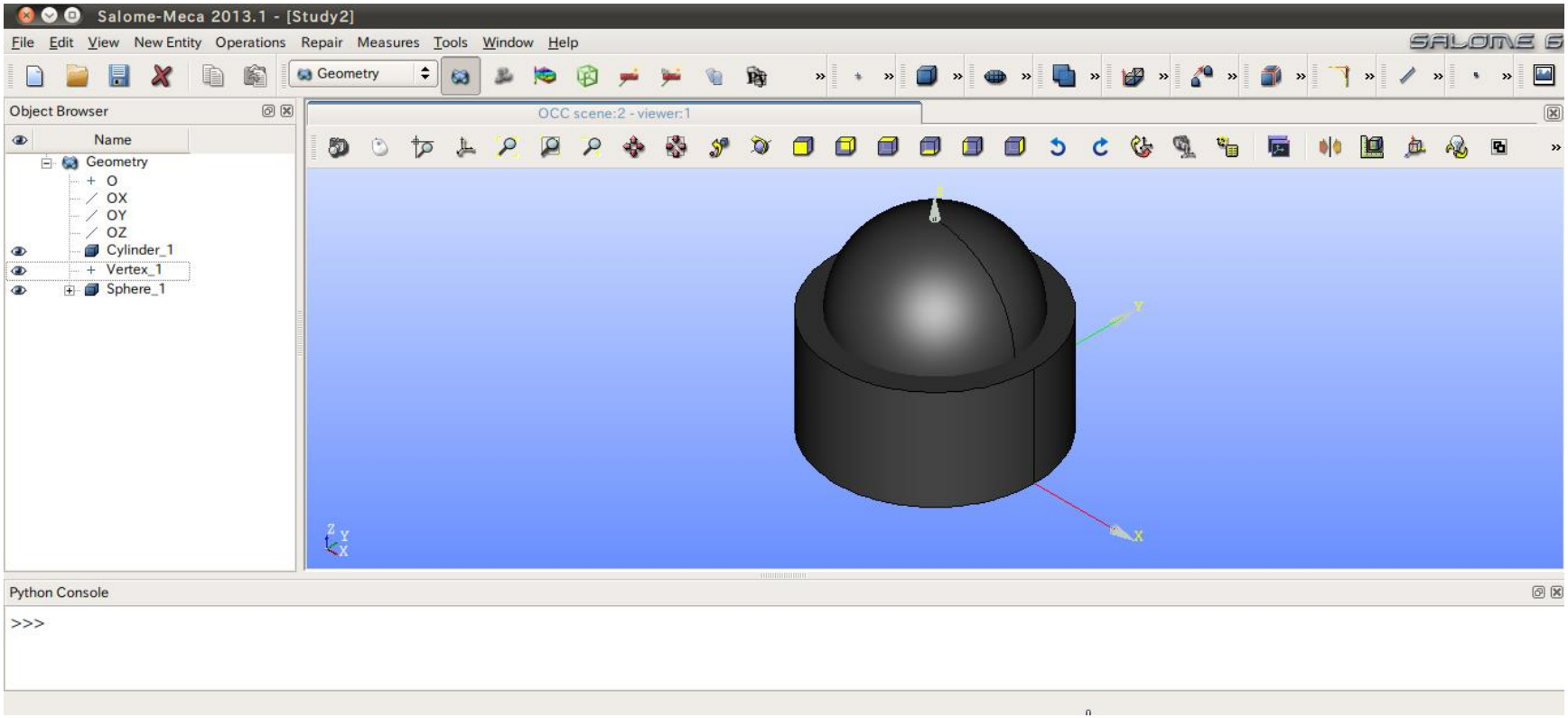
New Entity>Primitives>Sphere





矢印を選択するとグラフィックウインドウまたはオブジェクトブラウザから選択可能

演習1　Primitivesによるモデル作成

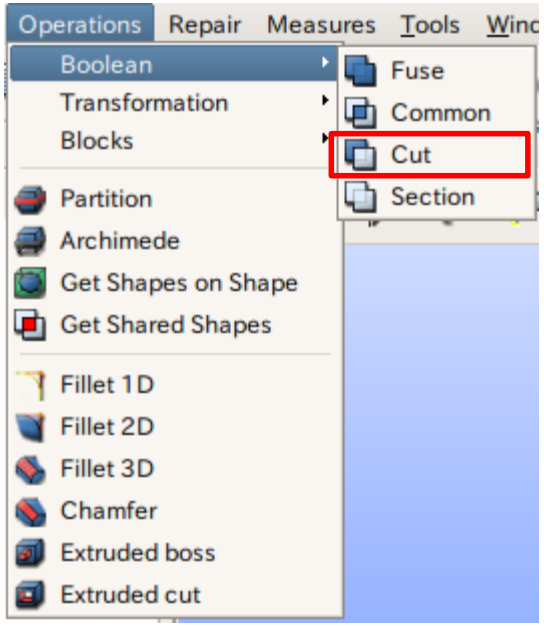


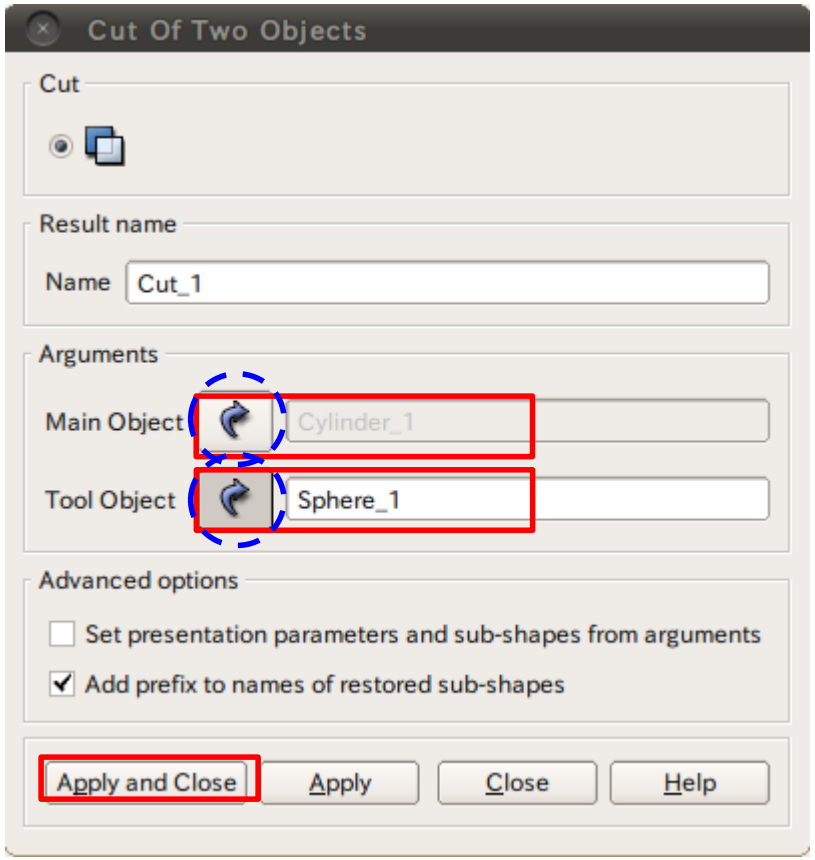
演習1　Primitivesによるモデル作成

② ソリッドモデルAに対し、点B（0，0，50）を中心点とする半径40mmの球形状を除去しなさい。これをソリッドモデルCとする。

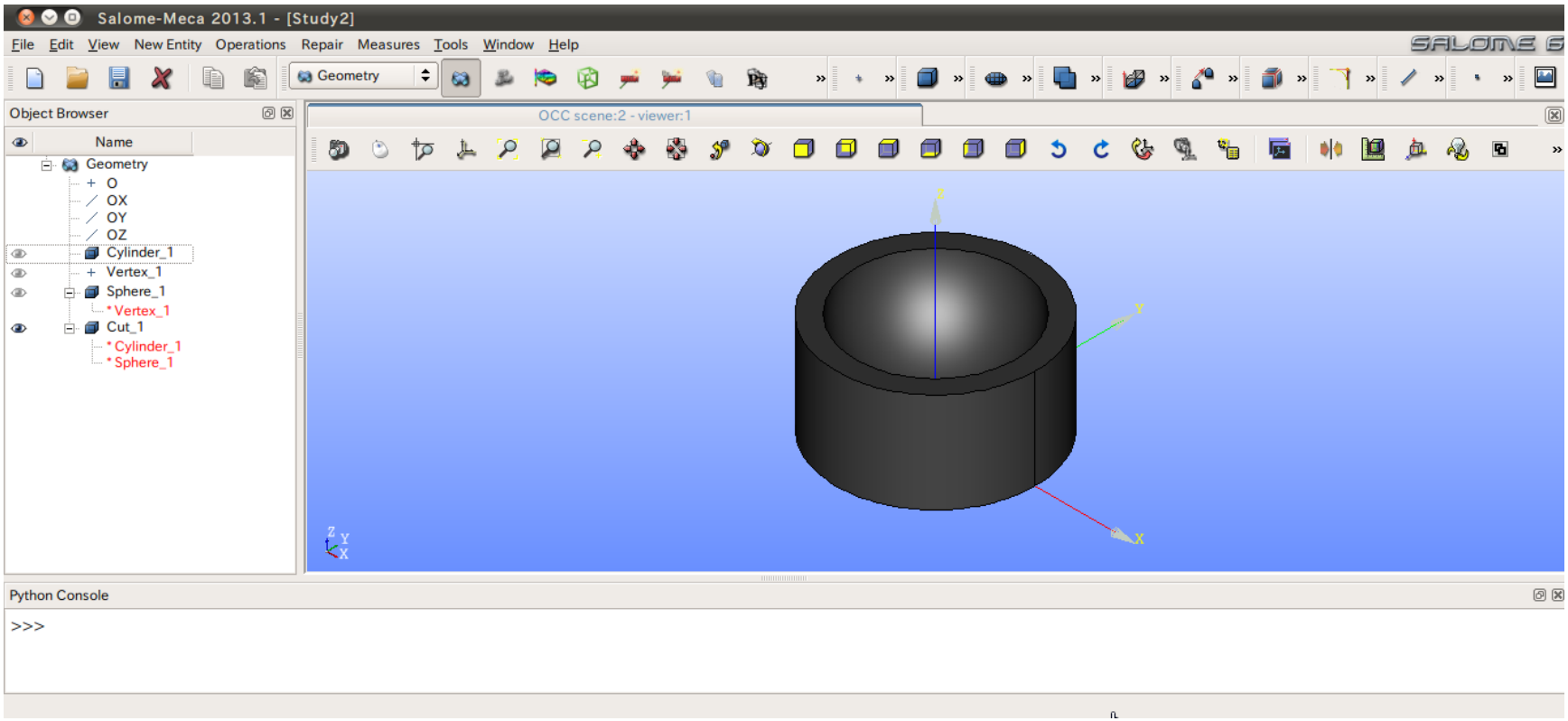
球の作成

Operations>Boolean>Cut





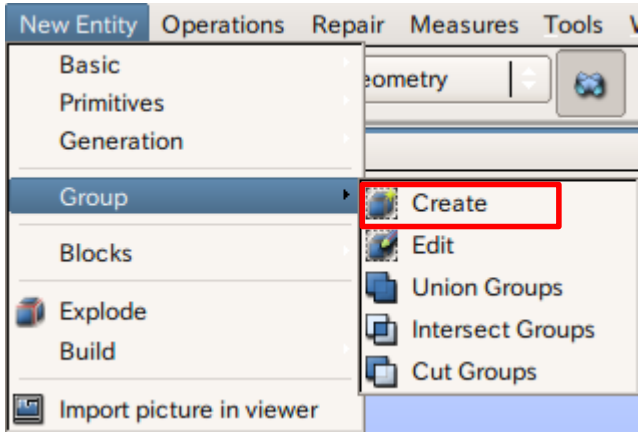
演習1　Primitivesによるモデル作成

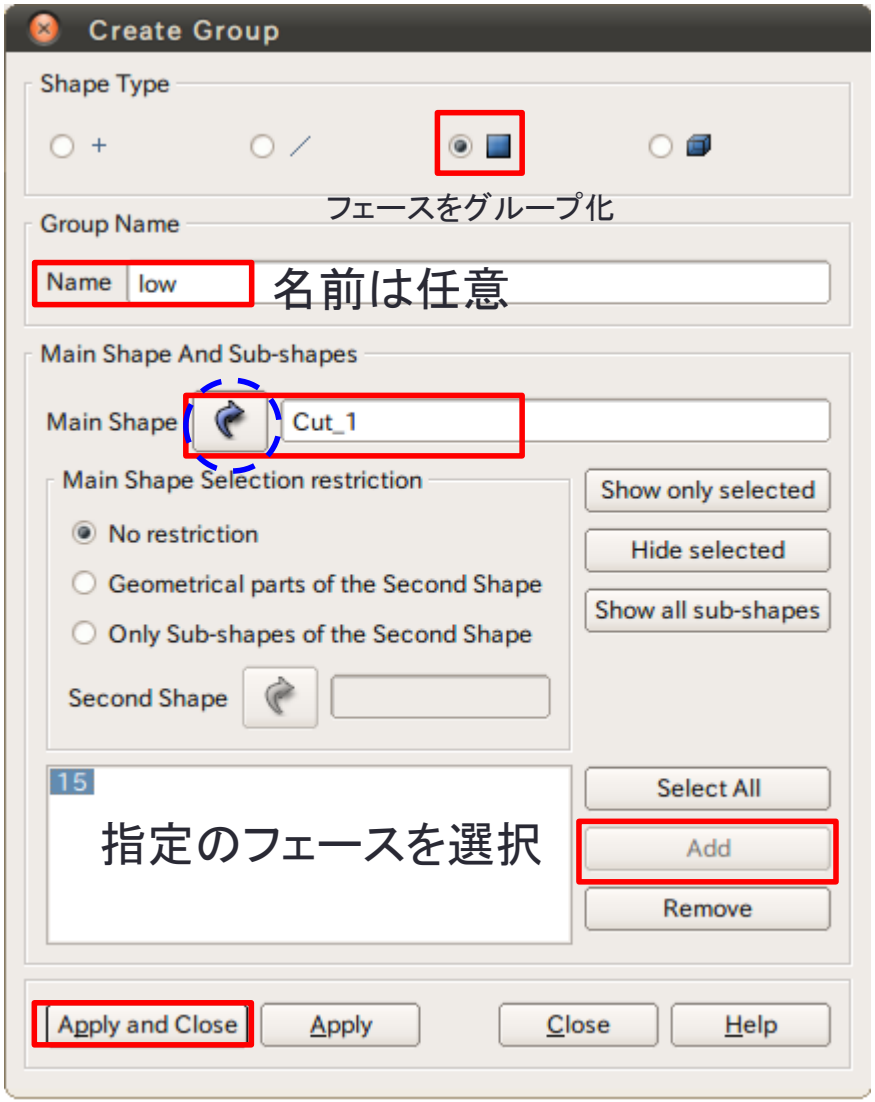


演習1　グループの作成

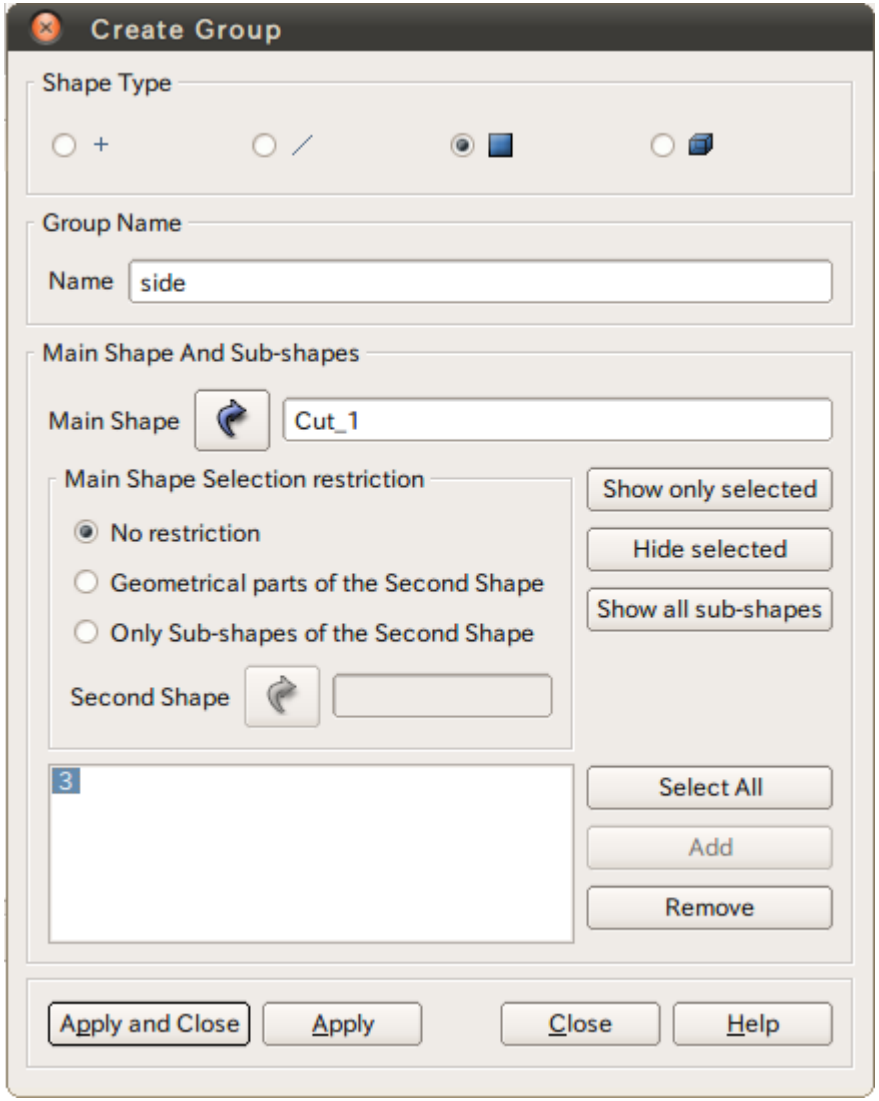
グループの作成

New Entity>Group>Create

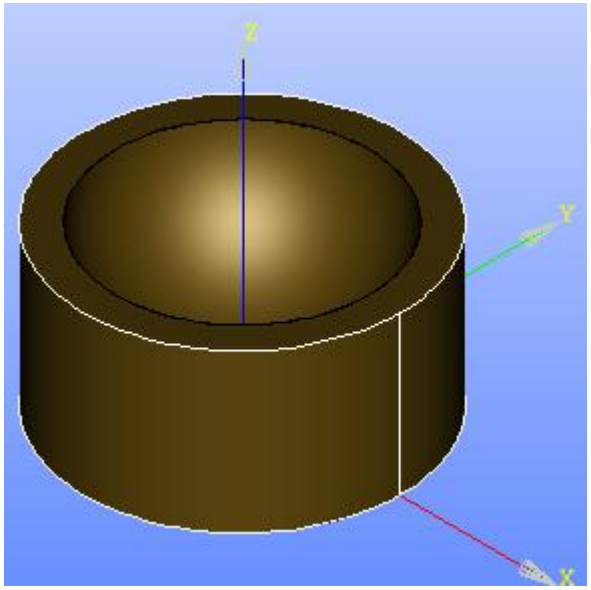




演習1　グループの作成



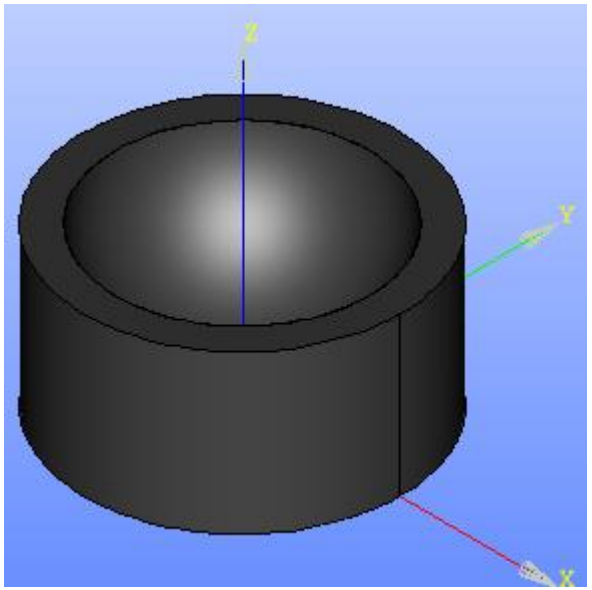
作成中

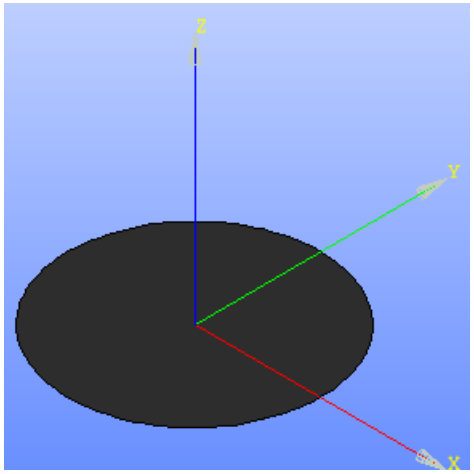


選択するとハイライトされる

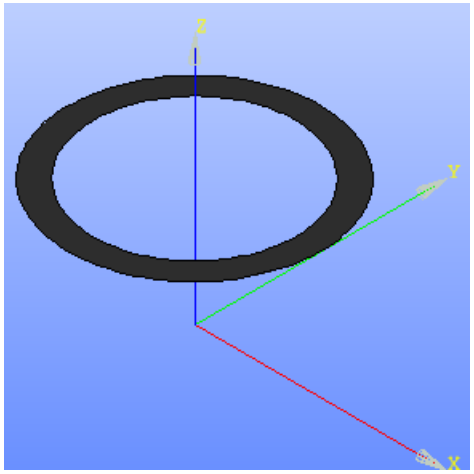
演習1　グループの作成

グループの作成

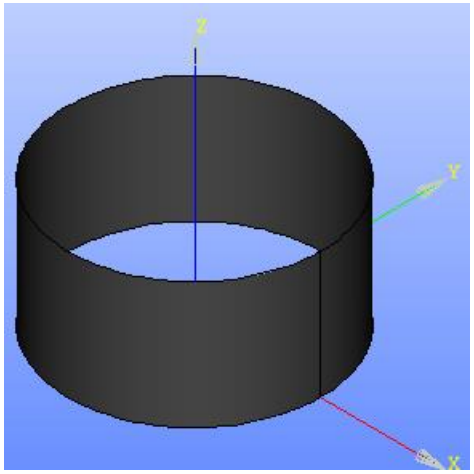




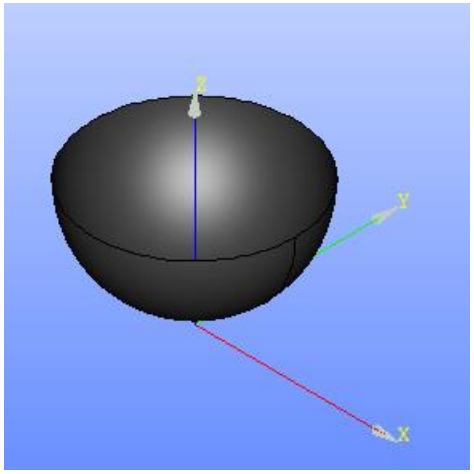
low



up



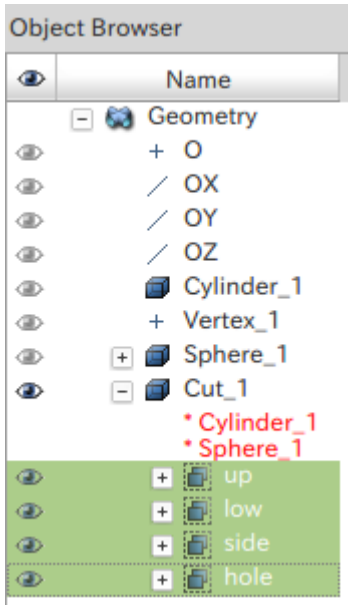
side



hole

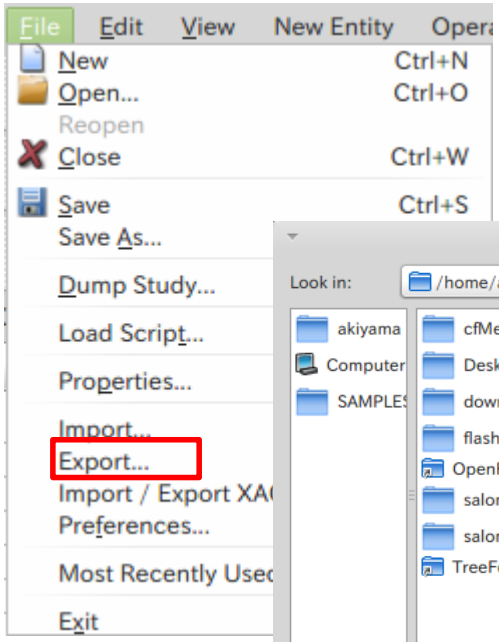
演習1　geometryの出力

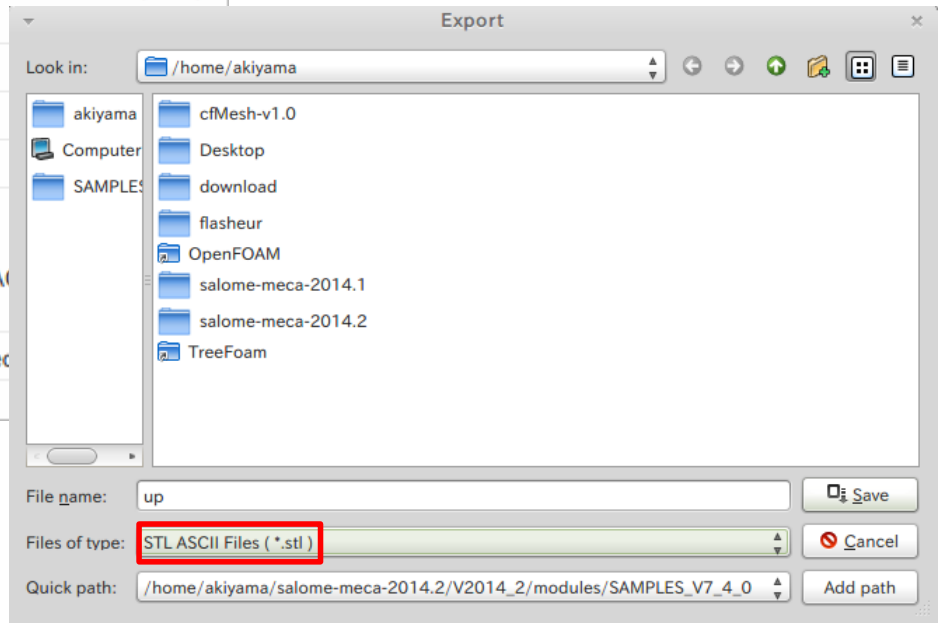
グループを選択した状態にする



geometryの出力

File>Export





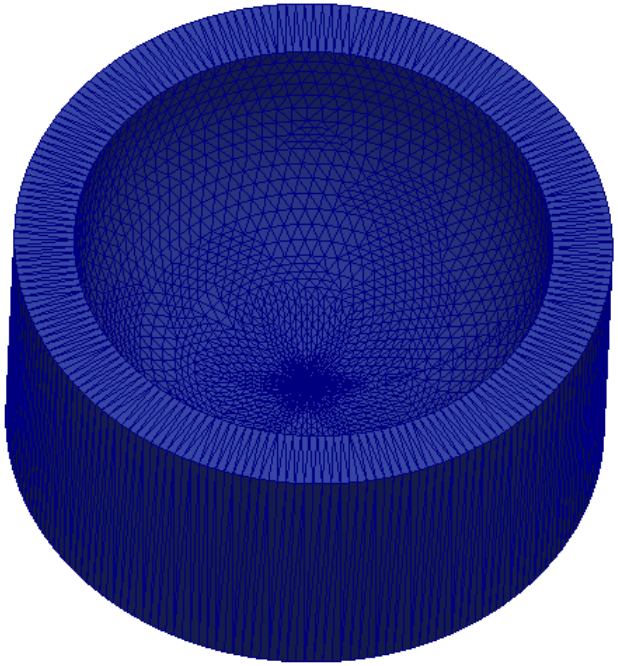
Cut_1と各サーフェスグループを出力

STL ASCII Filesを選択

演習1-1　Cut_1のcfMesh作成

system/mesh.Dict

```
/*--------------------------------*- C++ -*----------------------------------*¥
| =========                 |                                                 |
| ¥¥      /  F ield         | cfMesh: A library for mesh generation           |
|  ¥¥    /   O peration     |                                                 |
|   ¥¥  /    A nd           | Author: Franjo Juretic                          |
|    ¥¥/     M anipulation  | E-mail: franjo.juretic@c-fields.com             |
¥*---------------------------------------------------------------------------*/

FoamFile
{
    version   2.0;
    format    ascii;
    class     dictionary;
    location  "system";
    object    meshDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

surfaceFile        "Cut_1.stl";

maxCellSize        2;

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



Cut_1.stl

StlファイルとmaxCellSizeを指定するのみでメッシュ作成可能

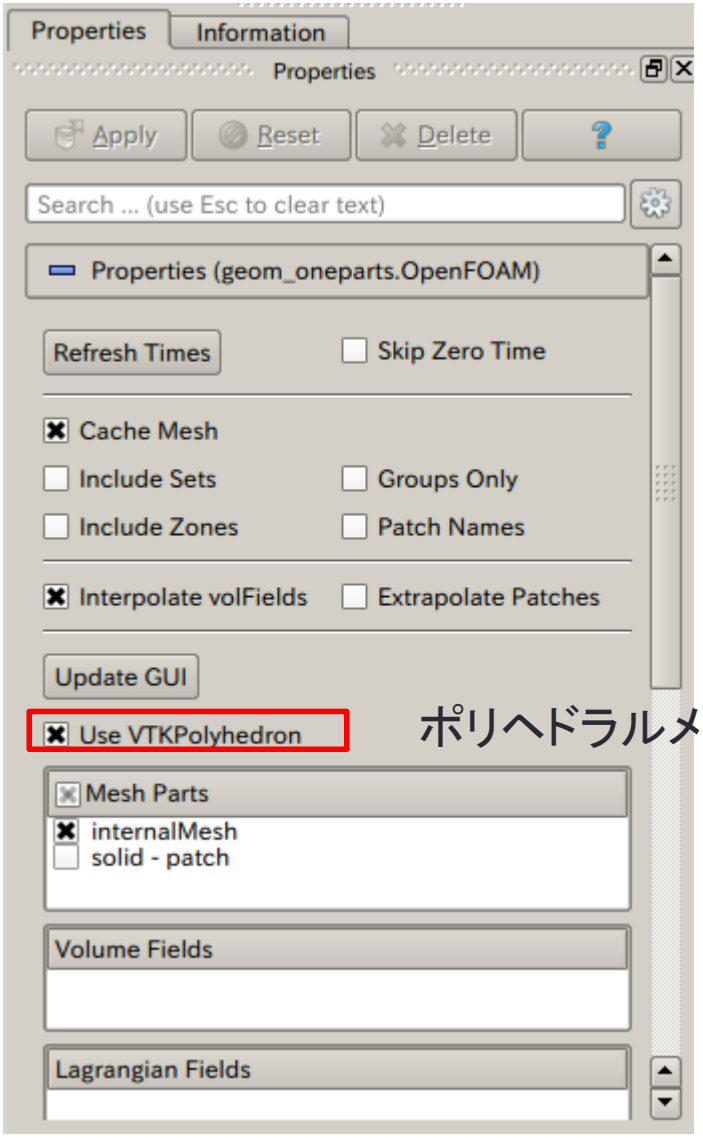
$cartesianMesh

演習1-1　Cut_1のcfMesh作成

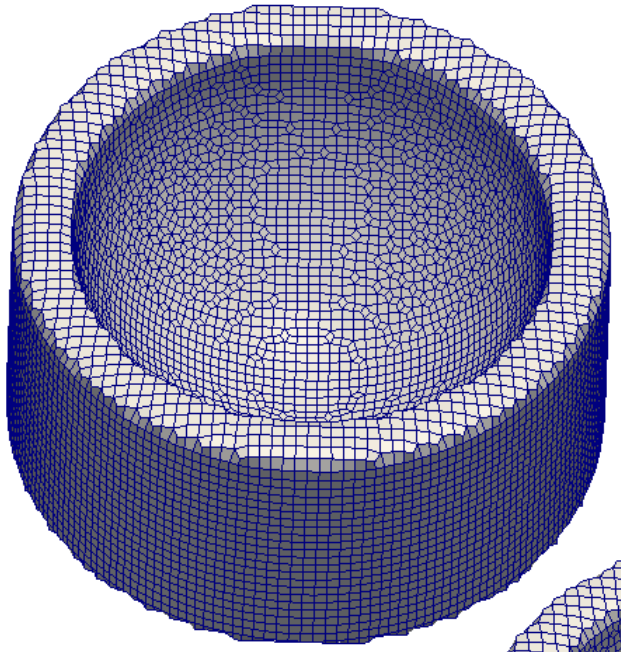
$checkMesh

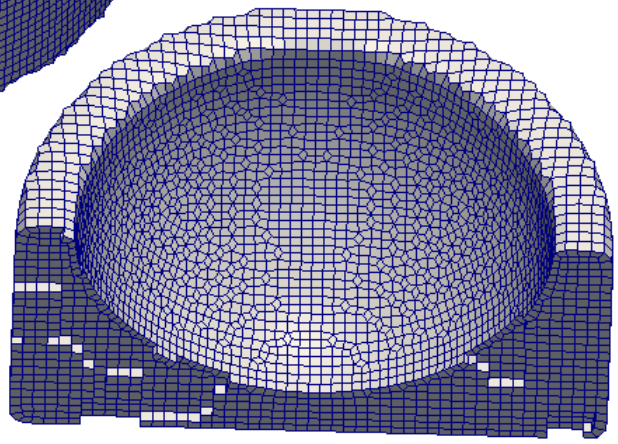
```
Create time

Create polyMesh for time = 0

Time = 0

Mesh stats
    points:           42707
    internal points:  31825
    faces:            116932
    internal faces:   106052
    cells:            37164
    faces per cell:   6
    boundary patches: 1
    point zones:      0
    face zones:       0
    cell zones:       0

Overall number of cells of each type:
    hexahedra:     37164
    prisms:        0
    wedges:        0
    pyramids:      0
    tet wedges:    0
    tetrahedra:    0
    polyhedra:     0

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
    Patch             Faces    Points   Surface topology
    solid             10880    10882    ok (closed singly connected)

Checking geometry...
    Overall domain bounding box (-49.9999 -49.9999 -3.76233e-28) (49.9998 49.9999 50)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Boundary openness (-5.16849e-17 3.7172e-17 -4.65867e-17) OK.
    Max cell openness = 3.31525e-16 OK.
    Max aspect ratio = 2.38403 OK.
    Minimum face area = 1.03424. Maximum face area = 5.4622.  Face area magnitudes OK.
    Min volume = 1.9871. Max volume = 10.035.  Total volume = 256821.  Cell volumes OK.
    Mesh non-orthogonality Max: 20.1503 average: 3.31061
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 0.493938 OK.
    Coupled point location match (average 0) OK.

Mesh OK.

End
```

演習1-1　Cut_1のcfMesh作成



ポリヘドラルメッシュの表示





特徴線が抽出されない

演習1-2　特徴線ありのcfMesh作成

Stlファイルから特徴線を抽出し*.fmsファイルへ変換する

$surfaceFeatureEdges -help

Usage: surfaceFeatureEdges [OPTIONS] <input surface file> <output surface file>
options:
  -angle <scalar> 特徴線の抽出角度
  -case <dir> specify alternate case directory, default is the cwd
  -noFunctionObjects
  do not execute functionObjects
  -srcDoc display source code in browser
  -doc display application documentation in browser
  -help print the usage

surfaceFeatureEdges -angle 10 Cut_1.stl Cut_1.fms

Stlファイル名　fmsファイル名

演習1-2　特徴線ありのcfMesh作成

meshDict内のCut_1.stlをCut_1.fmsに変更

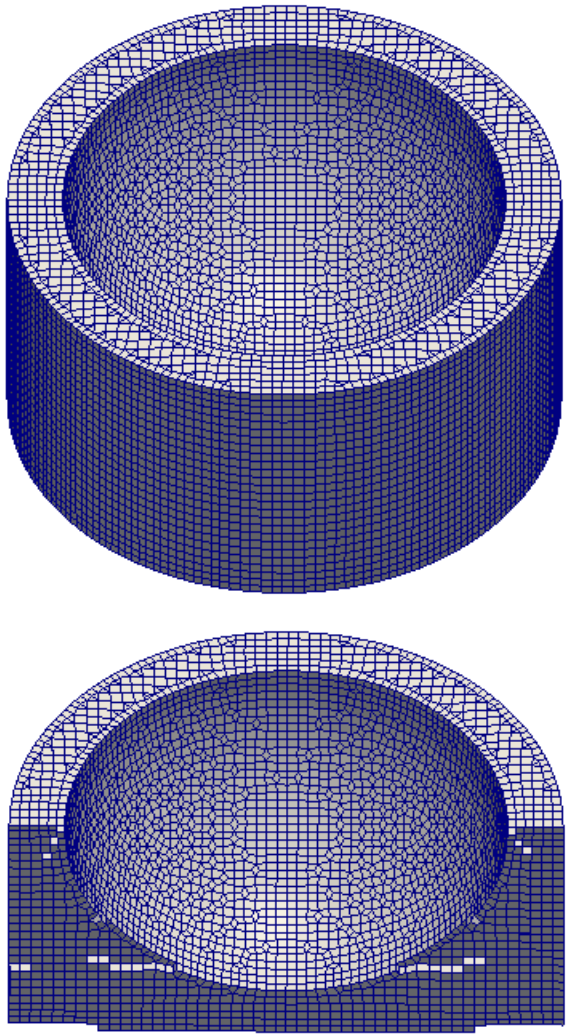
system/mesh.Dict

```
surfaceFile        "Cut_1.fms";

maxCellSize         2;
```

$checkMesh

Overall number of cells of each type:

hexahedra:	37428
prisms:	304
wedges:	0
pyramids:	760
tet wedges:	0
tetrahedra:	304
polyhedra:	0



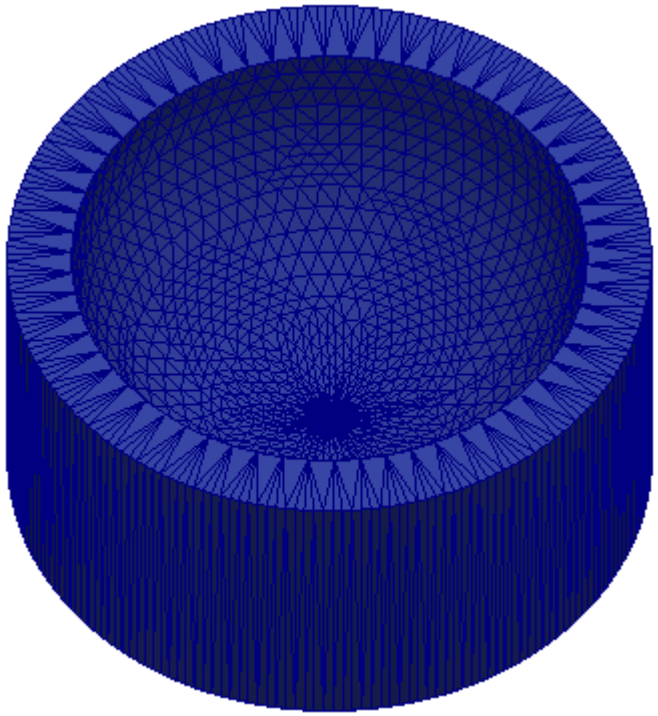
演習1-3　特徴線ありのcfMesh作成

オプションパラメータを追加
system/mesh.Dict

```
surfaceFile        "Cut_1.fms";

maxCellSize         2;

minCellSize          0.5;
```

--> FOAM FATAL ERROR:
Cannot construct the quadric surface for point (2.44921e-15 -5.99864e-31 10) because the normal does not exist!
This indicates that the input surface mesh is of poor quality

From function template<class ListType>
inline quadricFitting::quadricFitting(const point& origin, const vector normal, const ListType& otherPoints)
in file lnInclude/quadricFittingI.H at line 227.



Stlの品質が悪くcfMeshではメッシュ作成ができない

演習1-4　stlグループによるcfMesh作成

各サーフェスグループの*.stlファイルを修正する

```
solid
 facet normal -0.000000e+00 -0.000000e+00 -1.000000e+00
   outer loop
     vertex  4.823178e+01  1.317936e+01  0.000000e+00
     vertex  4.860115e+01  1.174430e+01  0.000000e+00
     vertex  4.736632e+01  1.601349e+01  0.000000e+00
   endloop
 endfacet
 facet normal -0.000000e+00 -0.000000e+00 -1.000000e+00
   outer loop
     vertex  4.782005e+01  1.460284e+01  0.000000e+00
     vertex  4.823178e+01  1.317936e+01  0.000000e+00
     vertex  4.736632e+01  1.601349e+01  0.000000e+00
   endloop
 endfacet
.
.
.
 facet normal  0.000000e+00  0.000000e+00 -1.000000e+00
   outer loop
     vertex  5.913659e+00  4.964905e+01  0.000000e+00
     vertex  1.174430e+01  4.860115e+01  0.000000e+00
     vertex  1.317936e+01  4.823178e+01  0.000000e+00
   endloop
 endfacet
endsolid
```

ファイル名を追加（patch名になる）

```
solid low
 facet normal -0.000000e+00 -0.000000e+00 -1.000000e+00
   outer loop
     vertex  4.823178e+01  1.317936e+01  0.000000e+00
     vertex  4.860115e+01  1.174430e+01  0.000000e+00
     vertex  4.736632e+01  1.601349e+01  0.000000e+00
   endloop
 endfacet
 facet normal -0.000000e+00 -0.000000e+00 -1.000000e+00
   outer loop
     vertex  4.782005e+01  1.460284e+01  0.000000e+00
     vertex  4.823178e+01  1.317936e+01  0.000000e+00
     vertex  4.736632e+01  1.601349e+01  0.000000e+00
   endloop
 endfacet
.
.
.
 facet normal  0.000000e+00  0.000000e+00 -1.000000e+00
   outer loop
     vertex  5.913659e+00  4.964905e+01  0.000000e+00
     vertex  1.174430e+01  4.860115e+01  0.000000e+00
     vertex  1.317936e+01  4.823178e+01  0.000000e+00
   endloop
 endfacet
endsolid
```

修正した*.stlファイルをマージする

STLファイルをマージするスクリプト

stl_merge

```
#!/bin/sh

files="*.stl"
for filepath in ${files}
do
  filename=`basename $filepath .stl`
sed -i -e "1s/solid/solid $filename/" $filename.stl

  echo $filename
done

cat *.stl > mesh.stl
```

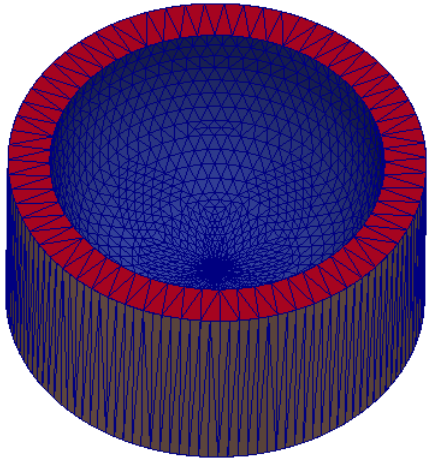
ディレクトリ内にあるstlファイルを検出
検出したstlファイルの数分ループを回す
Stlファイル名を取得
文字列”solid”を検索し”solid ファイル名”に置き換える

Stlファイルをマージし一つのmesh.stlファイルにする

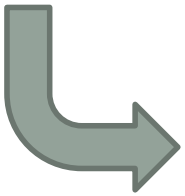
演習1-4　stlグループによるcfMesh作成

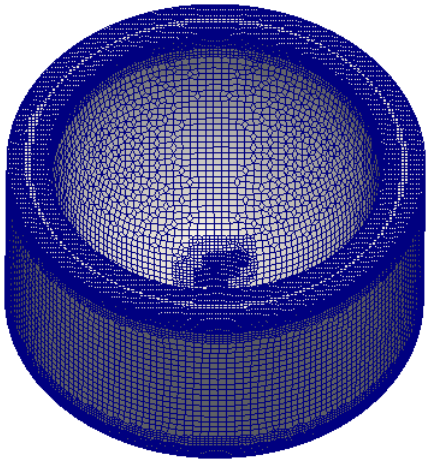
system/mesh.Dict

```
surfaceFile        "mesh.stl";

maxCellSize         2;

minCellSize          0.5;
```

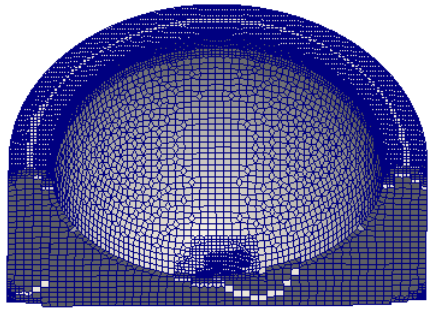


mesh.stl



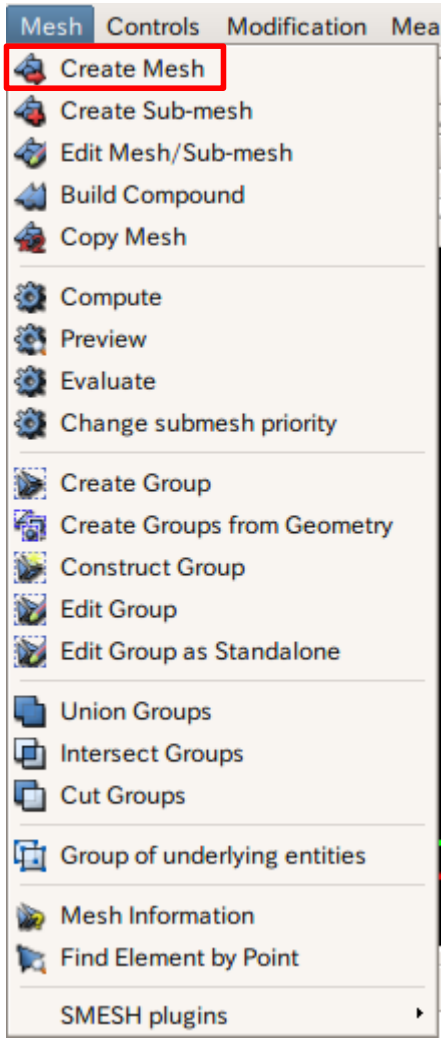


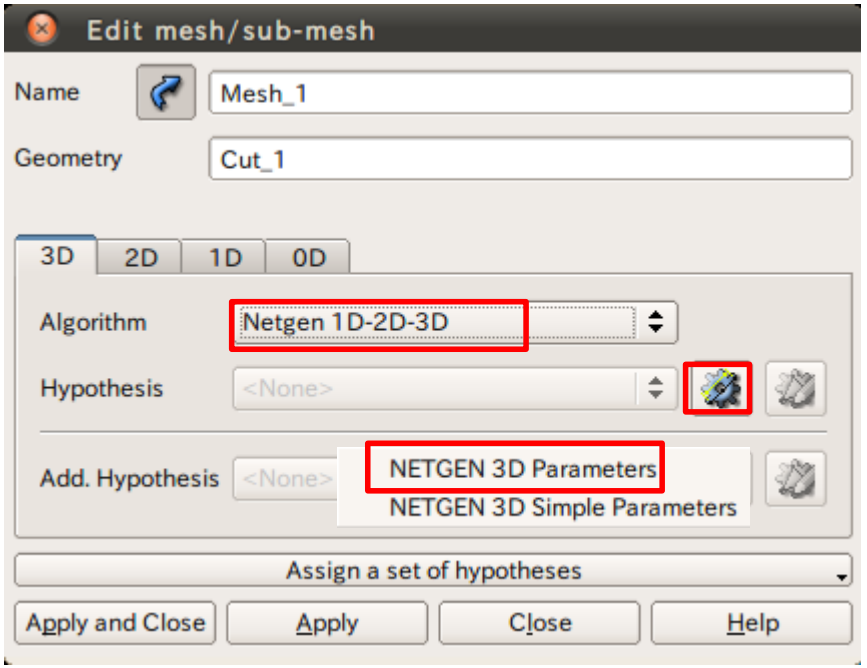
特徴線を抽出する（ surfaceFeatureEdges ）とメッシュ作成できないがstlでグループ分けをしておくとメッシュ作成できる



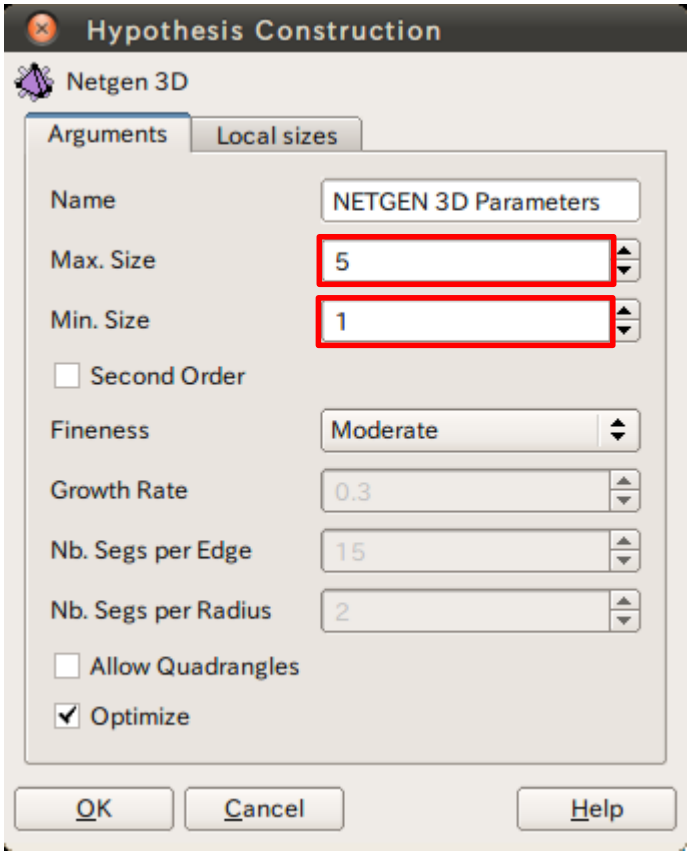
演習2　表面メッシュの作成

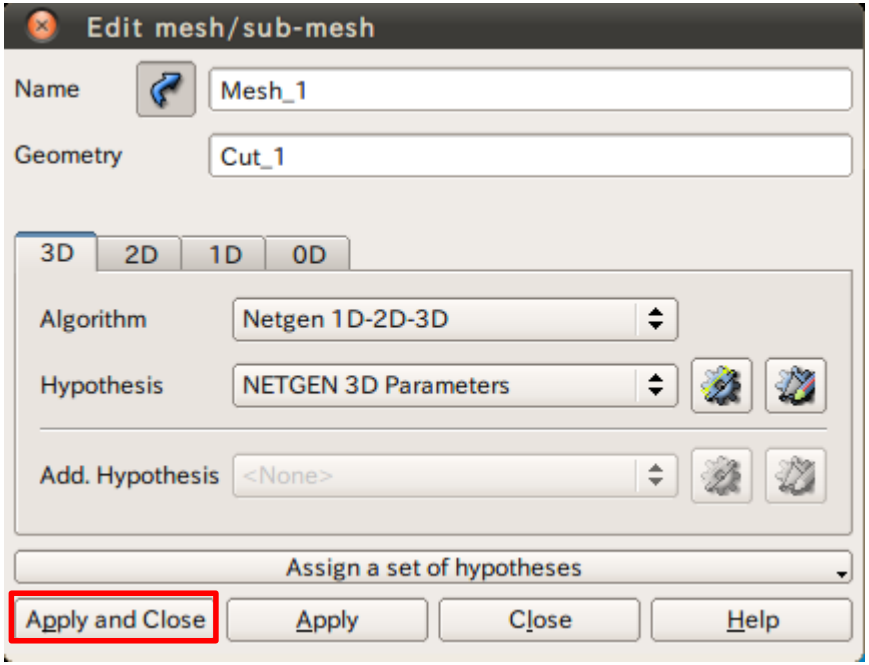
メッシュ設定
Mesh>Create Mesh





演習2　表面メッシュの作成





演習2　表面メッシュの作成

Mesh computation succeed

Compute mesh

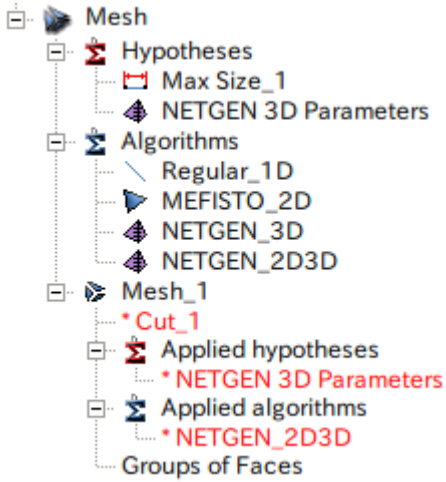
Name

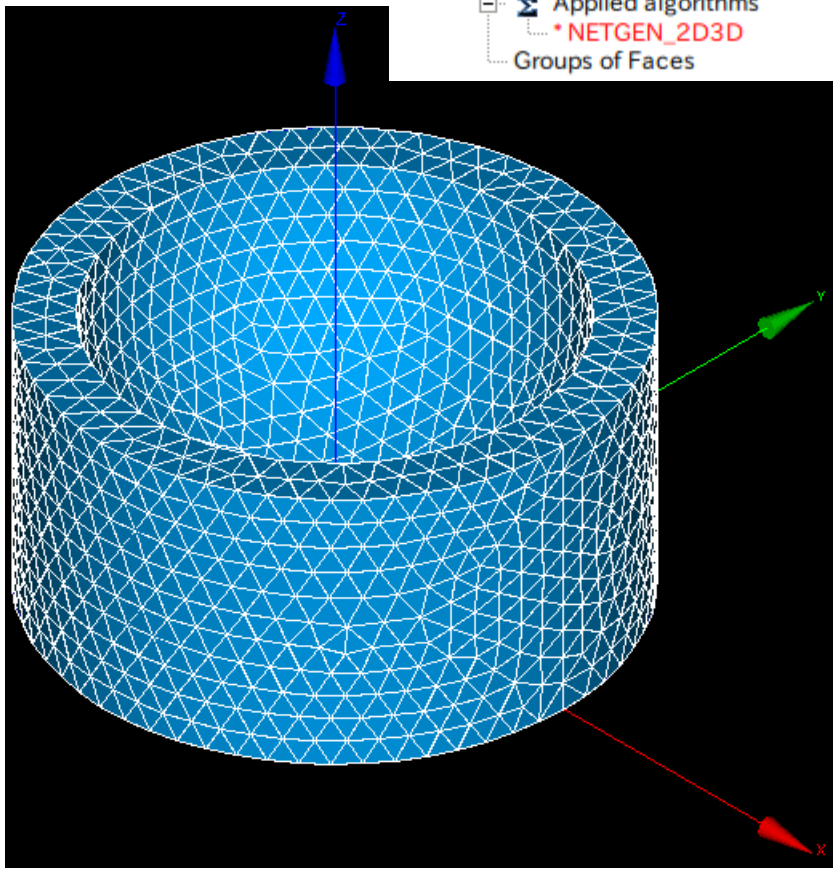
Mesh_1

Mesh Infos

	Total	*Linear*	*Quadratic*
Nodes :	2118		
0D Elements :	0		
Balls :	0		
Edges :	199	199	0
Faces :	3290	3290	0
Triangles :	3290	3290	0
Quadrangles :	0	0	0
Polygons :	0		
Volumes :	7441	7441	0
Tetrahedrons :	7441	7441	0
Hexahedrons :	0	0	0
Pyramids :	0	0	0
Prisms :	0	0	0
Hexagonal prisms :	0		
Polyhedrons :	0		

Close

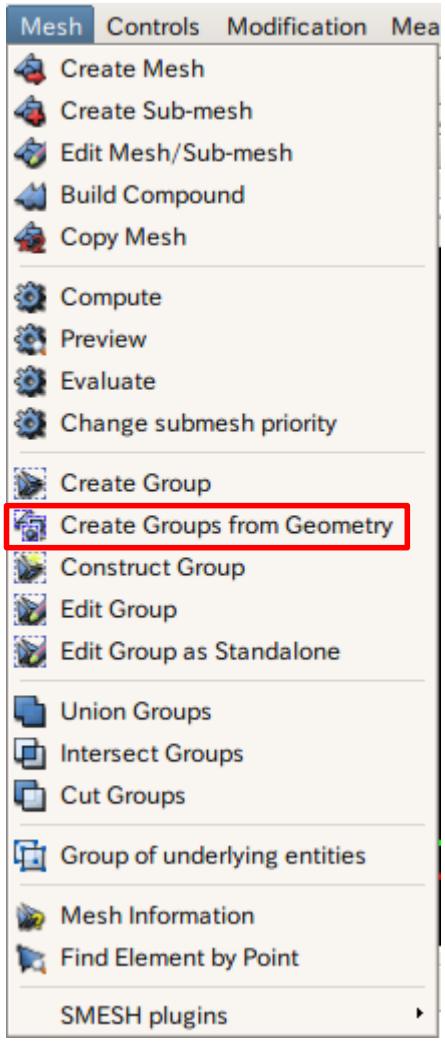


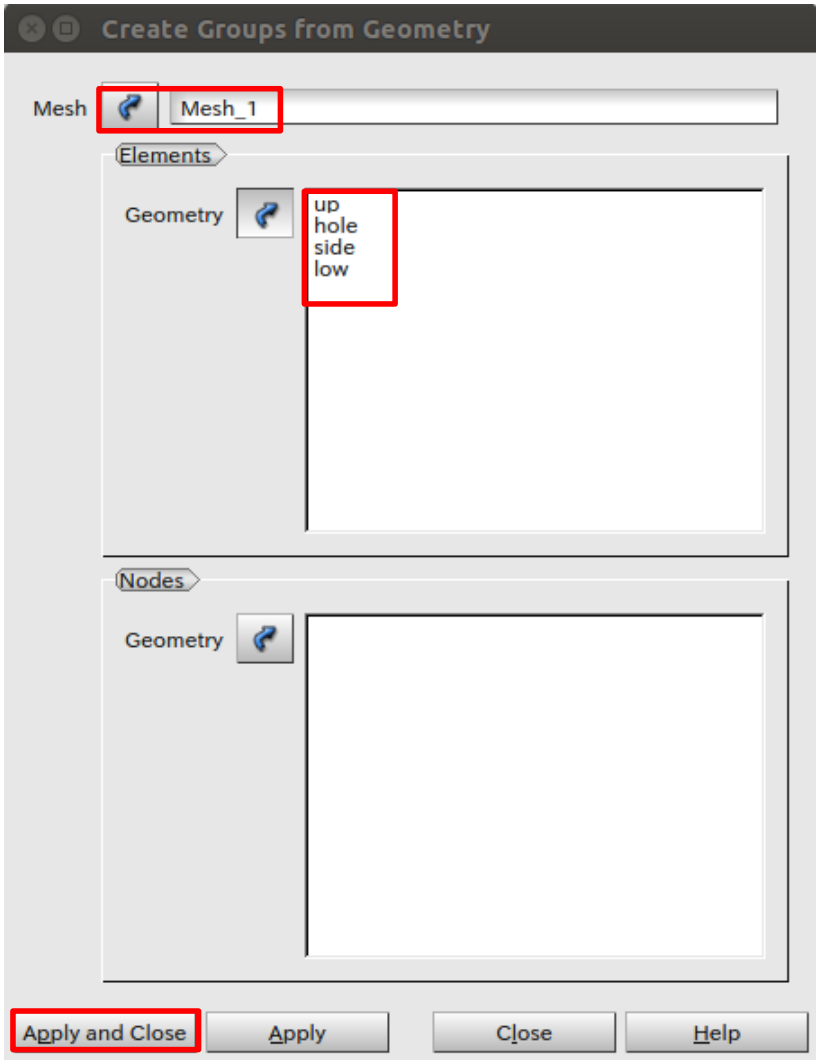


演習2　メッシュのグループ化

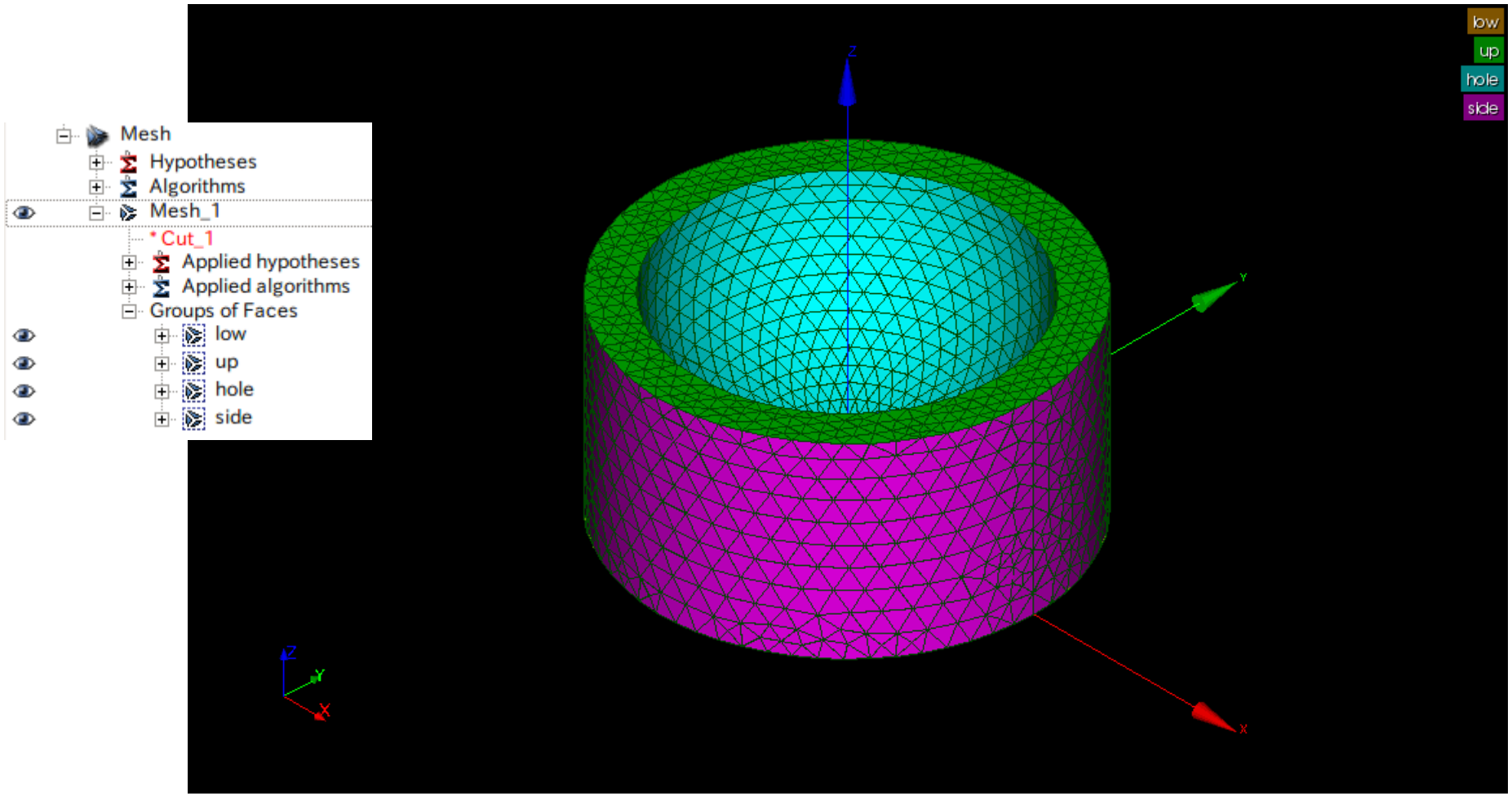
グループの作成

Mesh>Create Groups from Geometry



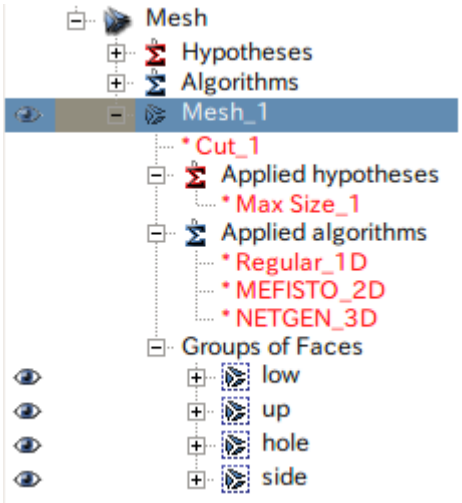


演習2　メッシュのグループ化



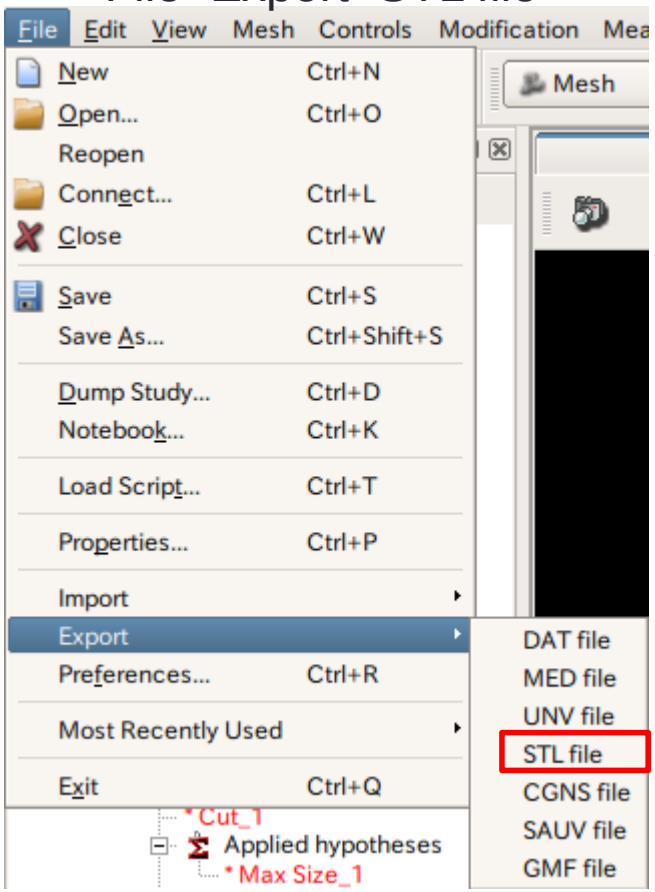
演習2　メッシュの出力

メッシュを選択した状態にする



メッシュの出力

File>Export>STL file



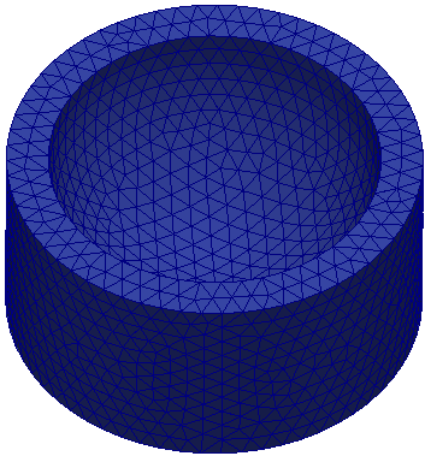
Mesh_1と各サーフェスグループを出力

演習2　Mesh_1のcfMesh作成

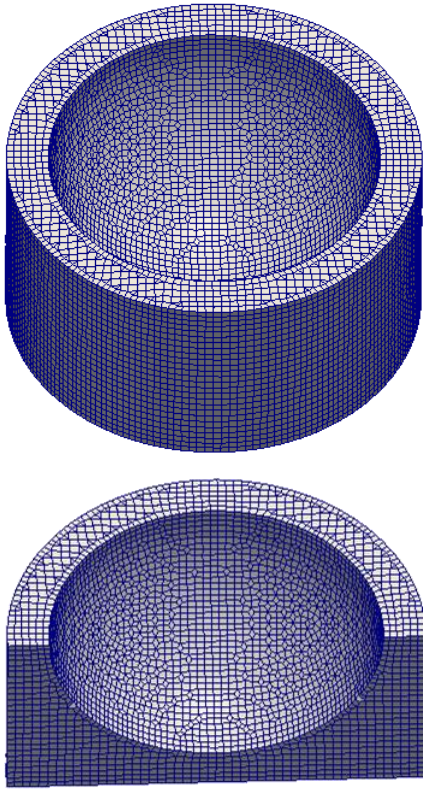
$surfaceFeatureEdges -angle 10 Mesh_1.stl Mesh_1.fms
$cartesianMesh

system/mesh.Dict

```
surfaceFile        “Mesh_1.fms";

maxCellSize         2;

minCellSize          0.5;
```



Mesh_1.stl

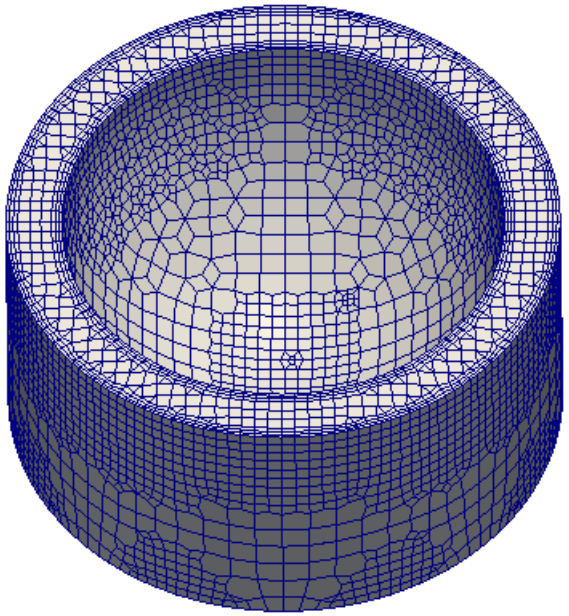


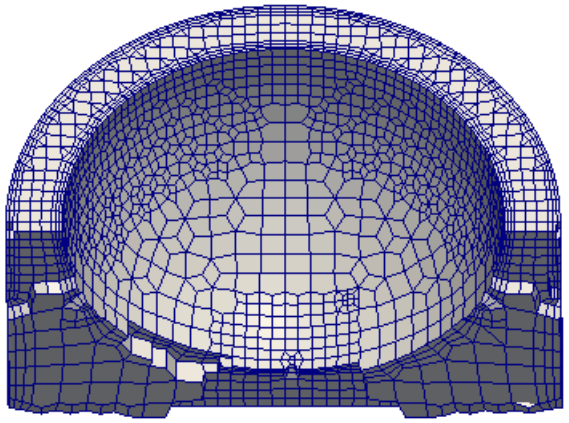
演習3-1　境界層の作成

```
surfaceFile        "mesh.fms";

maxCellSize        10;

minCellSize 2.5;

boundaryLayers
{

    patchBoundaryLayers
    {

        side
        {
            maxFirstLayerThickness        10;
            nLayers        3;
            thicknessRatio        1.2;
        }

        hole
        {
            maxFirstLayerThickness        10;
            nLayers        3;
            thicknessRatio        1.2;
        }
    }
}
```

表面メッシュを作成した各フェイスのstlファイルをマージ

$surfaceFeatureEdges -angle 10 mesh.stl mesh.fms
$cartesianMesh





meshDict

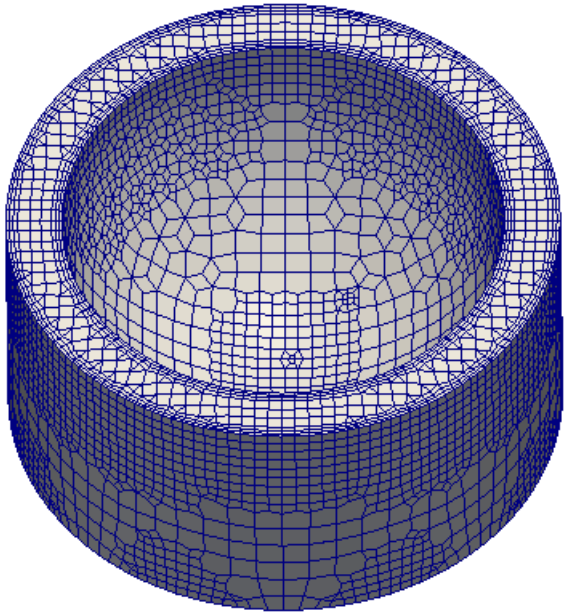
```
surfaceFile     "mesh.fms";

maxCellSize     10;

minCellSize 2.5;

boundaryLayers
{

    patchBoundaryLayers
    {

        side
        {
            maxFirstLayerThickness      10;
            nLayers     3;
            thicknessRatio      1.2;
        }

        hole
        {
            maxFirstLayerThickness      10;
            nLayers     3;
            thicknessRatio      1.2;
        }
    }
}
```

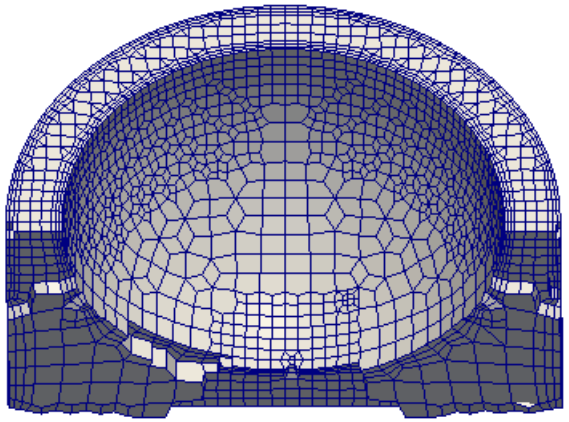
fmsファイル名

最大セルサイズ

最小セルサイズ

境界層第1層の最大サイズ
層数
成長率



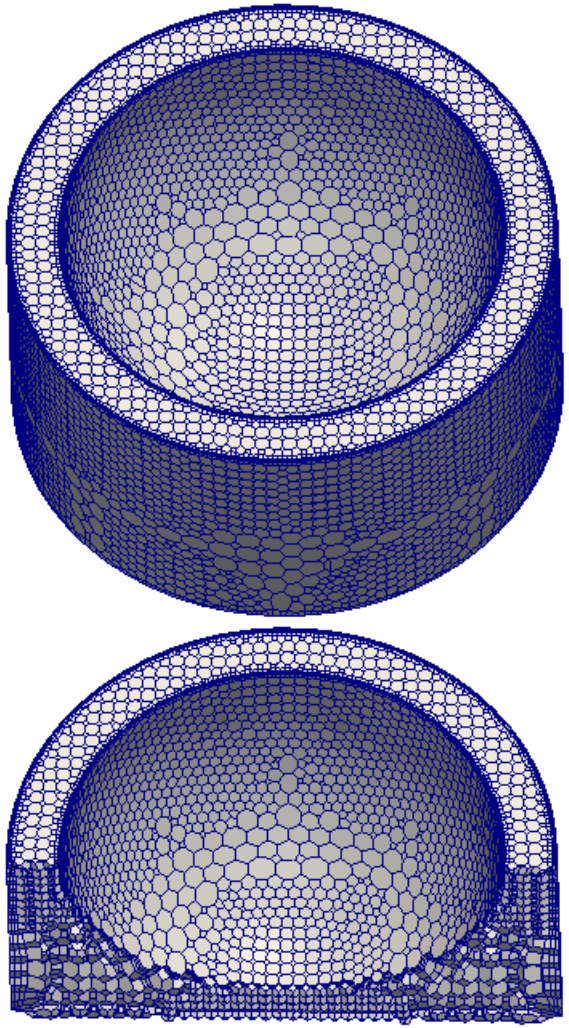


演習3-2　境界層の作成（ポリヘドラル）

3-1をコピーしpMeshを実行

$pMesh

```
surfaceFile        "mesh.fms";

maxCellSize        10;

minCellSize 2.5;

boundaryLayers
{

    patchBoundaryLayers
    {

        side
        {
            maxFirstLayerThickness        10;
            nLayers        3;
            thicknessRatio        1.2;
        }

        hole
        {
            maxFirstLayerThickness        10;
            nLayers        3;
            thicknessRatio        1.2;
        }
    }
}
```



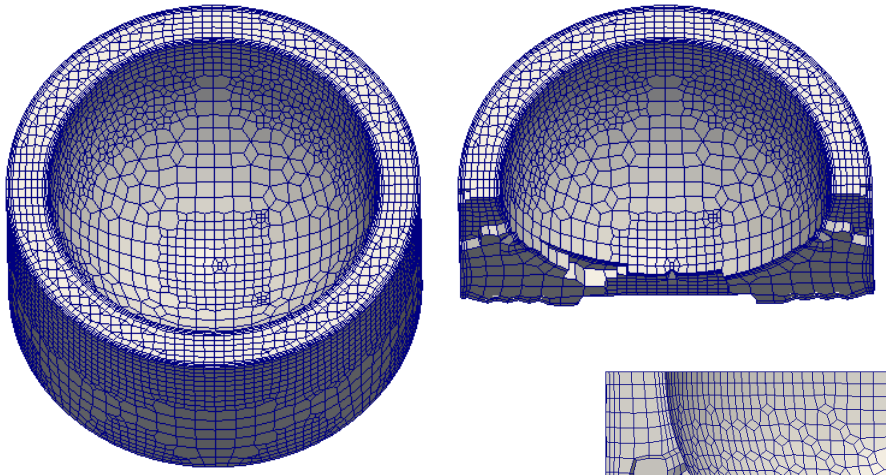
演習3-3　境界層の作成（スムーズ）

```
surfaceFile        "mesh.fms";
maxCellSize        10;
minCellSize 2.5;
boundaryLayers
{
    patchBoundaryLayers
    {
        side
        {
            maxFirstLayerThickness        10;
            nLayers        3;
            thicknessRatio        1.2;
        }
        hole
        {
            maxFirstLayerThickness        10;
            nLayers        3;
            thicknessRatio        1.2;
        }
    }
    optimiseLayer 1;
    optimisationParameters
    {
        nSmoothNormals 3;
        maxNumIterations 5;
        featureSizeFactor 0.4;
        reCalculateNormals 1;
        relThicknessTol 0.1;
    }
}
```

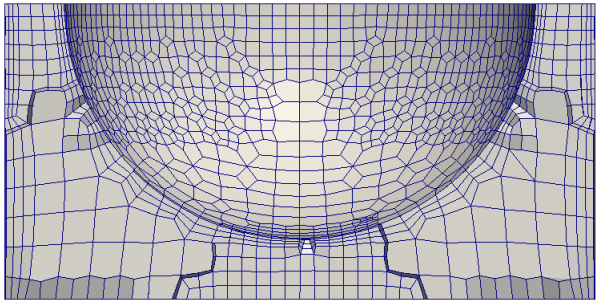
boundaryLayers 内に
スムーズオプションを追加

3-1をコピーしcartesianMeshを実行

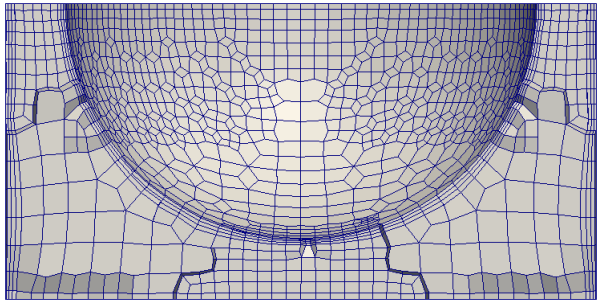
$cartesianMesh



演習3-1

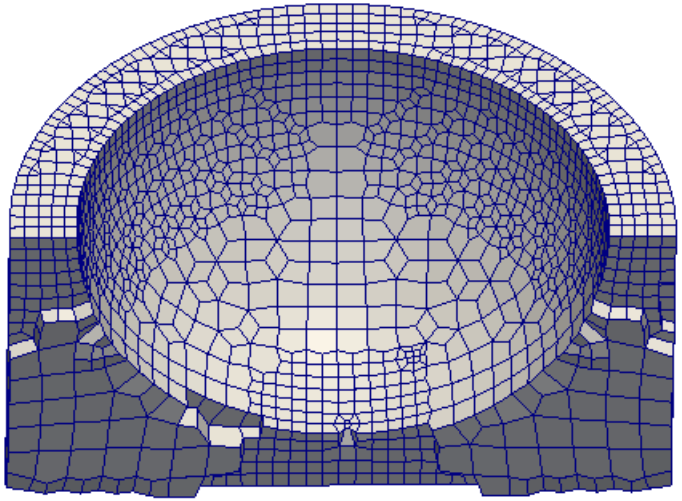


演習3-3

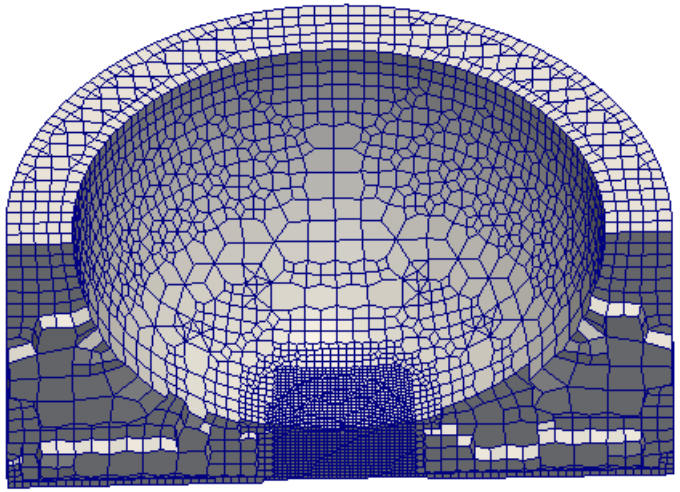


演習4　部分的なセルサイズ指定

```
surfaceFile        "mesh.fms";

maxCellSize        10;

minCellSize 2.5;

localRefinement
{

    low
    {
        cellSize 2;
//       additionalRefinementLevels 1;
    }
}

objectRefinements
{
    area1
    {
        cellSize 1.25;
        type box;
        centre  (0 0 0);
        lengthX  20;
        lengthY  20;
        lengthZ  20;
    }
}
```

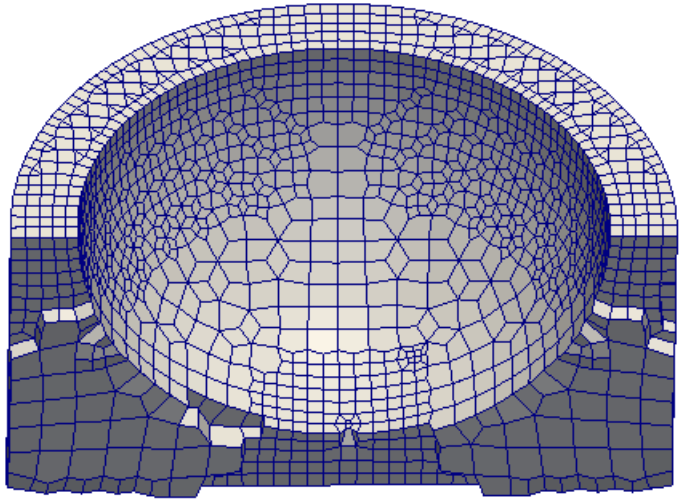


localRefinement, objectRefinements無し

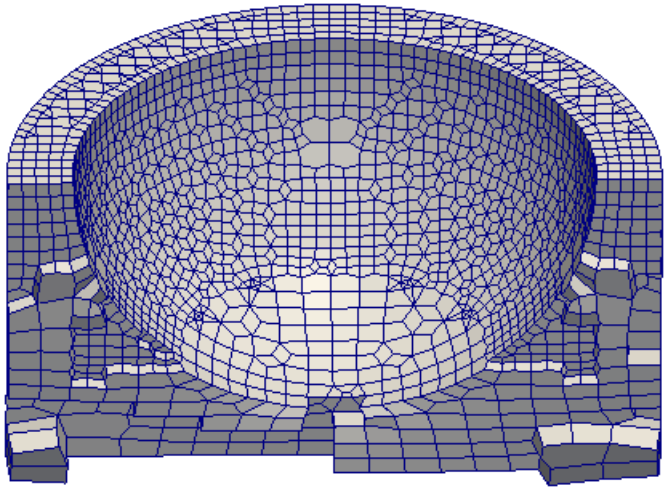


演習5　異方性メッシュの作成

```
surfaceFile        "mesh.fms";

maxCellSize        10;

minCellSize 2.5;

anisotropicSources
{
    Box
    {
        type box;
        centre  (0 0 0);
        lengthX  40;
        lengthY  40;
        lengthZ  40;
        scaleX 1;
        scaleY 1;
        scaleZ 0.5;
    }
}
```

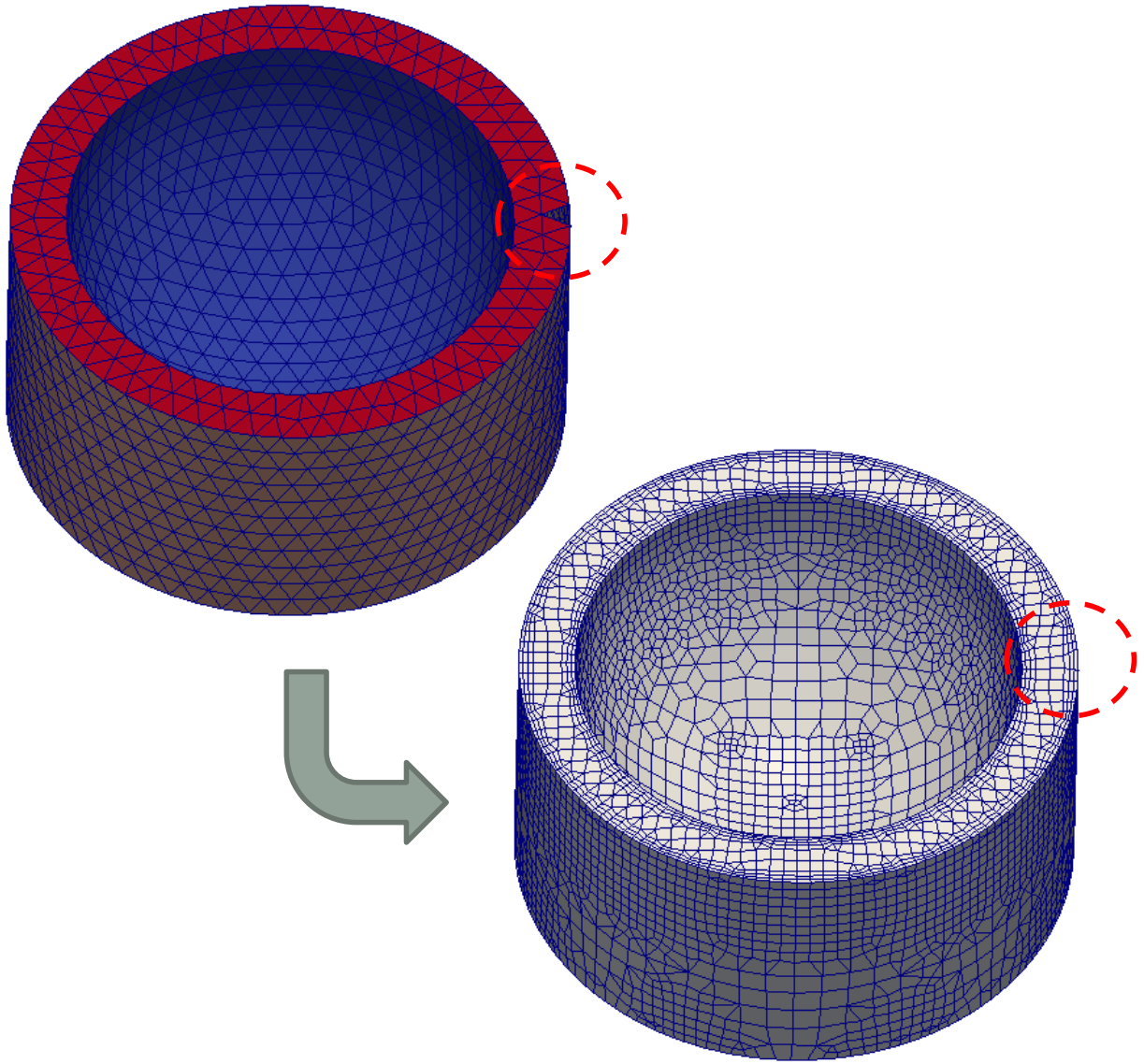


anisotropicSources無し



演習6　欠けた形状のメッシュ作成

```
surfaceFile        "mesh.fms";

maxCellSize        10;

minCellSize 2.5;

boundaryLayers
{
        maxFirstLayerThickness        10;
        nLayers        3;
        thicknessRatio        1.2;
}
```



演習7　snappyHexMeshとの比較

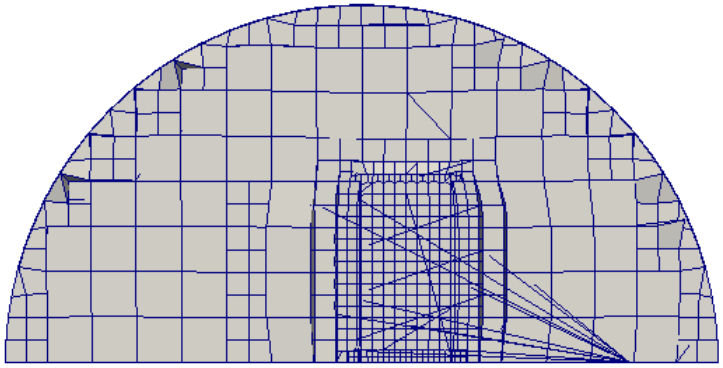
iglooWithFridgesをsnappyHexMeshとcfMeshで比較する

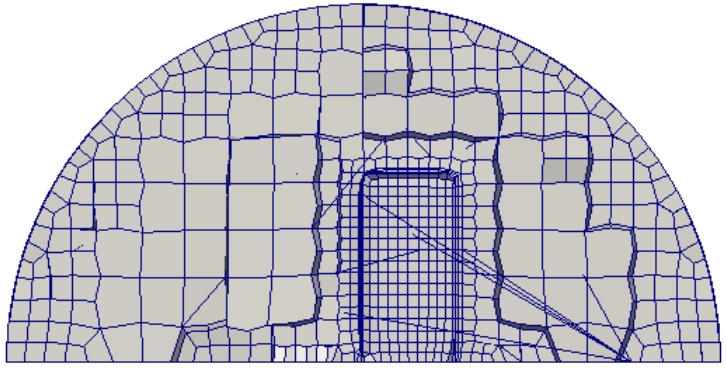
チュートリアル内のiglooWithFridgesを./Allrunで実行
表面メッシュを抽出
$foamToSurface -latestTime mesh.stl

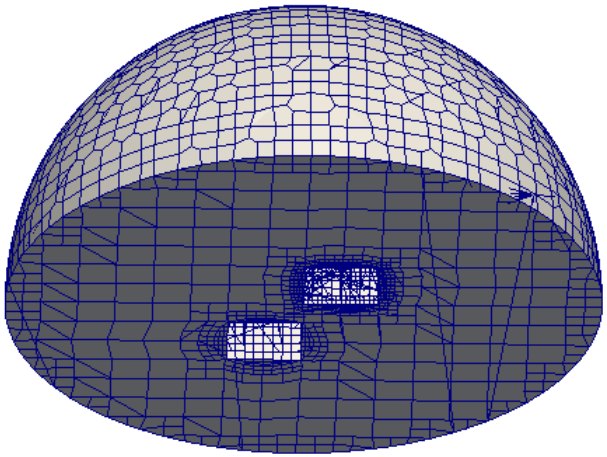
```
maxCellSize        0.5;
surfaceFile        "mesh.stl";
boundaryLayers
{
    patchBoundaryLayers
    {
        twoFridgeFreezers_seal_0
        {
            maxFirstLayerThickness        0.1;
            nLayers        3;
            thicknessRatio        1.2;
        }
        twoFridgeFreezers_herring_1
        {
            maxFirstLayerThickness        0.1;
            nLayers        3;
            thicknessRatio        1.2;
        }
    }
}
```

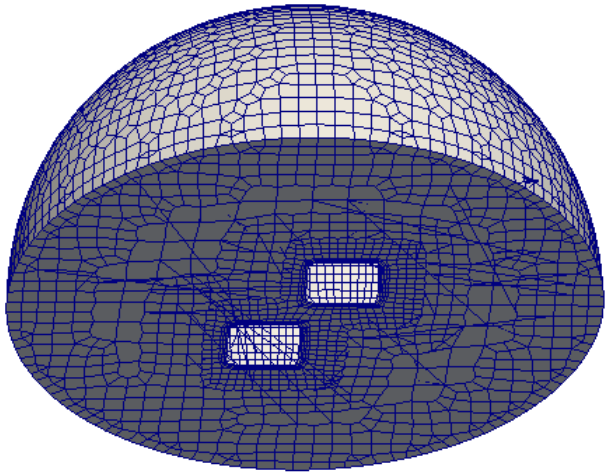
```
localRefinement
{
    igloo
    {
        cellSize        0.25;
//        additionalRefinementLevels        1;
    }
        twoFridgeFreezers_seal_0
    {
        cellSize        0.125;
//        additionalRefinementLevels        2;
    }
        twoFridgeFreezers_herring_1
    {
        cellSize        0.125;
//        additionalRefinementLevels        2;
    }
}
```

演習7　snappyHexMeshとの比較









snappyHexMesh

cfMesh

演習7　snappyHexMeshとの比較

Mesh stats
- points: 14257
- faces: 33693
- internal faces: 30195
- cells: 9998
- faces per cell: 6.39008
- boundary patches: 9
- point zones: 0
- face zones: 0
- cell zones: 0

Overall number of cells of each type:
- hexahedra: 6911
- prisms: 208
- wedges: 0
- pyramids: 0
- tet wedges: 12
- tetrahedra: 0
- polyhedra: 2867

Breakdown of polyhedra by number of faces:

faces	number of cells
4	397
5	168
6	702
7	362
8	40
9	962
11	15
12	99
14	1

snappyHexMesh

Mesh stats
- points: 17644
- faces: 45510
- internal faces: 40456
- cells: 13942
- faces per cell: 6.16597
- boundary patches: 4
- point zones: 0
- face zones: 0
- cell zones: 0

Overall number of cells of each type:
- hexahedra: 12503
- prisms: 128
- wedges: 0
- pyramids: 280
- tet wedges: 0
- tetrahedra: 112
- polyhedra: 919

Breakdown of polyhedra by number of faces:

faces	number of cells
6	176
7	71
8	20
9	458
12	114
15	61
18	19

cfMesh

演習7　snappyHexMeshとの比較

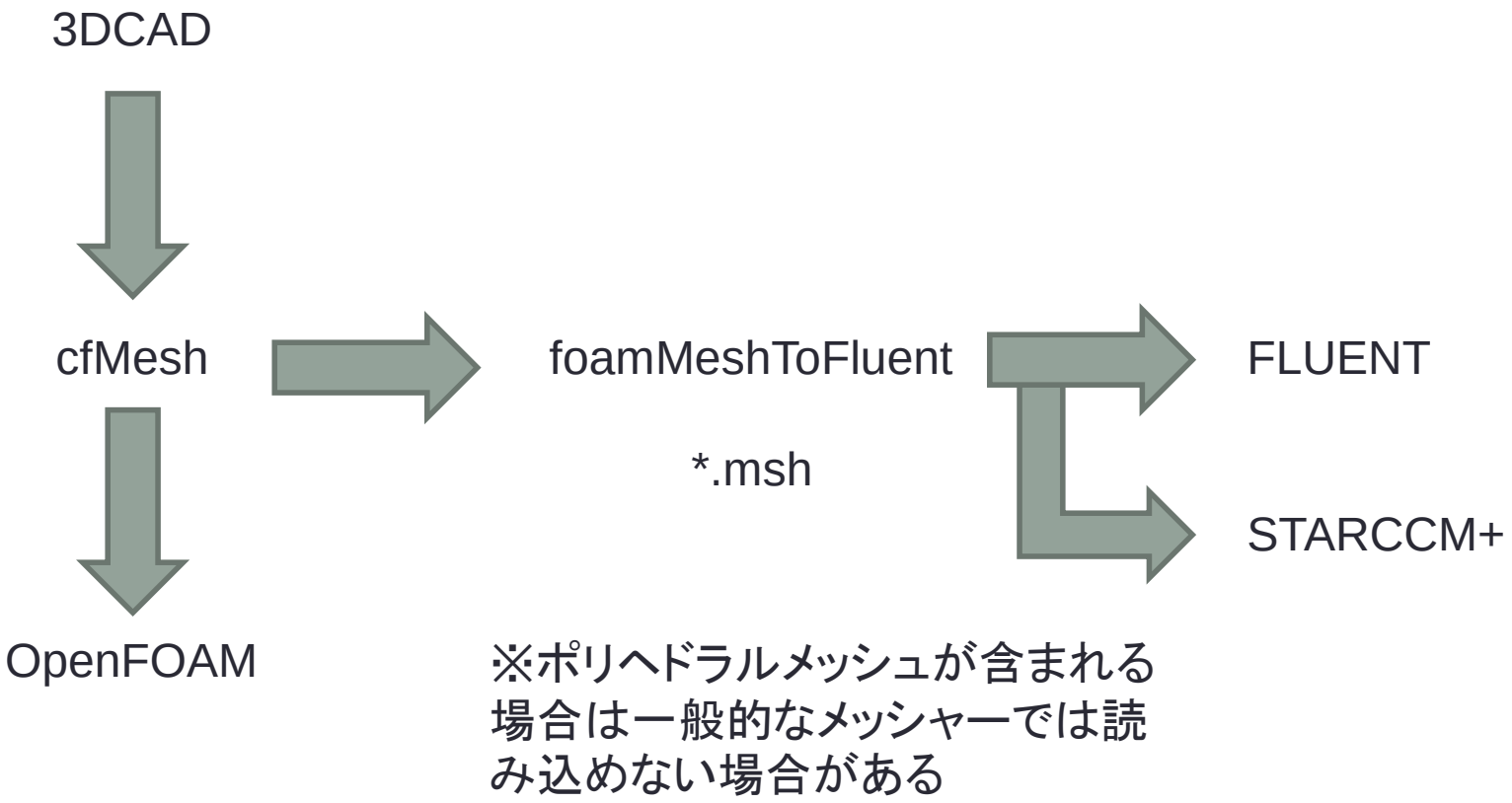
```
Checking patch topology for multiply connected surfaces...
                Patch    Faces   Points               Surface topology
                 maxY        0        0                  ok (empty)
                 minX        0        0                  ok (empty)
                 maxX        0        0                  ok (empty)
                 minY        0        0                  ok (empty)
               ground      830      993  ok (non-closed singly connected)
                 maxZ        0        0                  ok (empty)
               igloo     1276     1649  ok (non-closed singly connected)
   twoFridgeFreezers_seal_0      800      941  ok (non-closed singly
connected)
   twoFridgeFreezers_herring_1      592      625  ok (non-closed singly
connected)

Checking geometry...
    Overall domain bounding box (-1.00389 -1.00395 0) (7.00389 7.00395 4)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Boundary openness (3.52436e-17 5.92793e-17 -5.43521e-16) OK.
    Max cell openness = 3.20833e-16 OK.
    Max aspect ratio = 11.3868 OK.
    Minimum face area = 0.000490601. Maximum face area = 0.280257.
Face area magnitudes OK.
    Min volume = 2.1449e-05. Max volume = 0.143679.  Total volume =
129.524.  Cell volumes OK.
    Mesh non-orthogonality Max: 51.809 average: 14.8515
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 3.42668 OK.
    Coupled point location match (average 0) OK.

Mesh OK.
```

snappyHexMesh

```
Checking patch topology for multiply connected surfaces...
                Patch    Faces   Points               Surface topology
               ground     1326     1452  ok (non-closed singly connected)
               igloo     2228     2273  ok (non-closed singly connected)
   twoFridgeFreezers_seal_0      750      769  ok (non-closed singly connected)
   twoFridgeFreezers_herring_1      750      769  ok (non-closed singly
connected)

Checking geometry...
    Overall domain bounding box (-1.00386 -1.00391 0) (7.00386 7.00391
3.99987)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Boundary openness (-5.93818e-17 -1.84773e-16 3.63569e-16) OK.
    Max cell openness = 2.50477e-16 OK.
    Max aspect ratio = 11.0631 OK.
    Minimum face area = 0.000290858. Maximum face area = 0.307806.  Face
area magnitudes OK.
    Min volume = 2.93306e-05. Max volume = 0.175793.  Total volume =
129.341.  Cell volumes OK.
    Mesh non-orthogonality Max: 44.2681 average: 8.02187
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 2.1956 OK.
    Coupled point location match (average 0) OK.

Mesh OK.
```

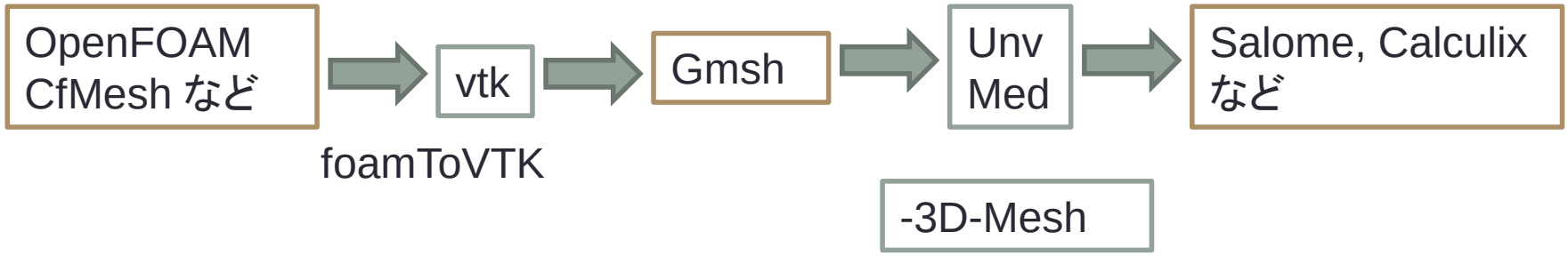
cfMesh

商用ソルバーへの変換


3DCAD
cfMesh
OpenFOAM
foamMeshToFluent
*.msh
FLUENT
STARCCM+
※ポリヘドラルメッシュが含まれる場合は一般的なメッシャーでは読み込めない場合がある

CfMesh の構造解析へのメッシュ変換

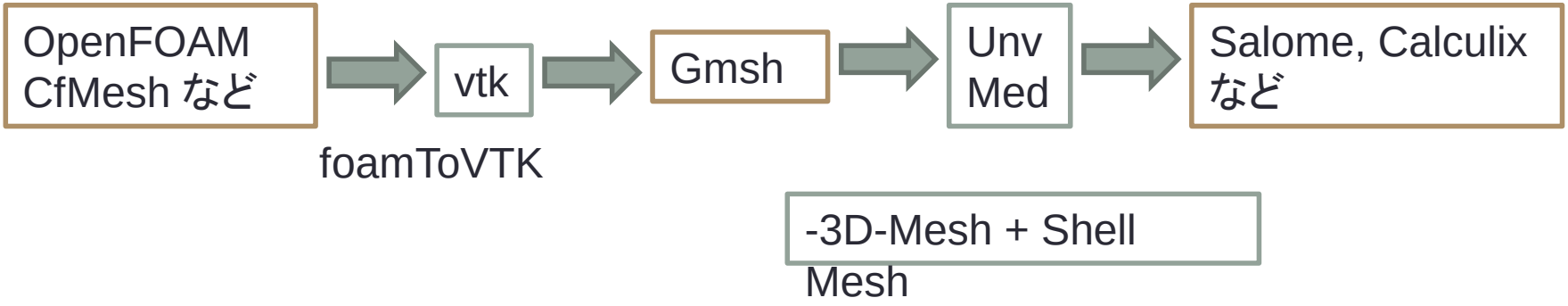
- OpenFOAMのメッシュは”foamToVTK” でVTKファイルに変換できるので、これをGmsh で読み込んで、Universal 形式か Med 形式で出力する



http://opencae.gifu-nct.ac.jp/pukiwiki/index.php?plugin=attach&pcmd=open&file=OpenCAE2014-09-20_SH-pptx.pdf&refer=%C2%E8%A3%B3%A3%B4%B2%F3%CA%D9%B6%AF%B2%F1%A1%A7H260920

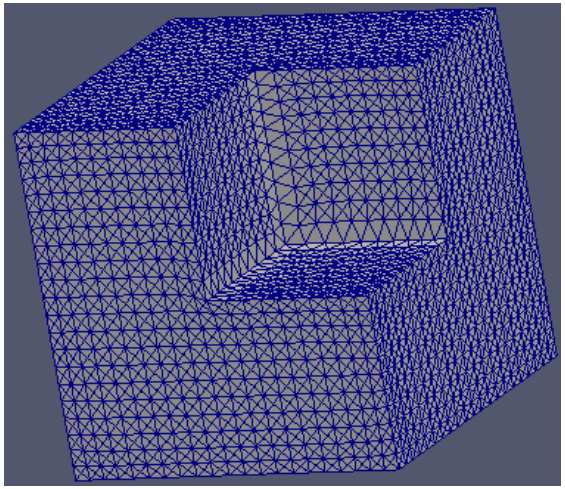
CfMesh の構造解析へのメッシュ変換

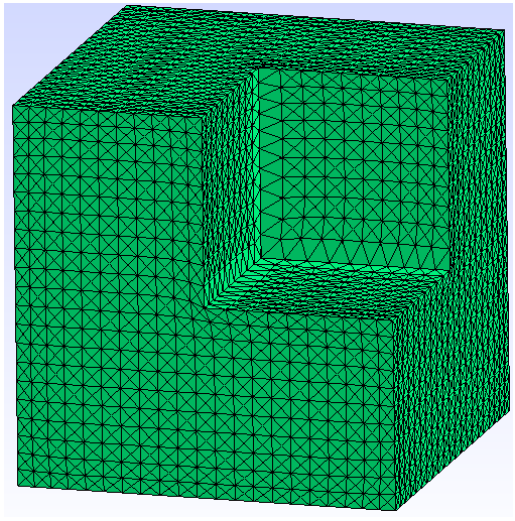
- OpenFOAMでメッシュを作成する際に面グループに名前を付けておくと、この面のグループは3次元のメッシュとは別に表面メッシュ(Shell 要素) として”foamToVTK” でVTKファイルに変換できる



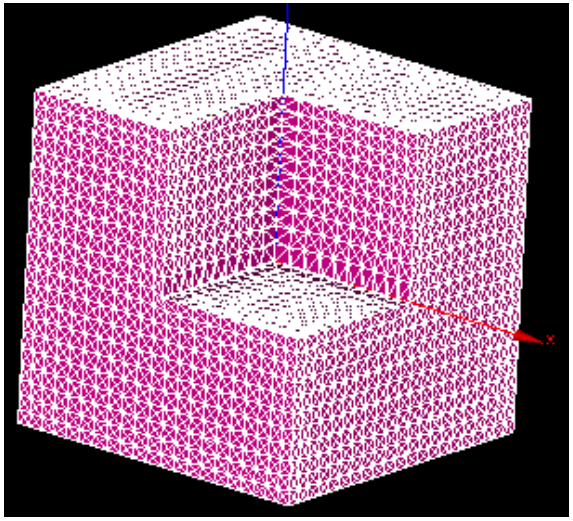
http://opencae.gifu-nct.ac.jp/pukiwiki/index.php?plugin=attach&pcmd=open&file=OpenCAE2014-09-20_SH-pptx.pdf&refer=%C2%E8%A3%B3%A3%B4%B2%F3%CA%D9%B6%AF%B2%F1%A1%A7H260920

cfMesh(tetMesh)で構造解析用メッシュを作成する

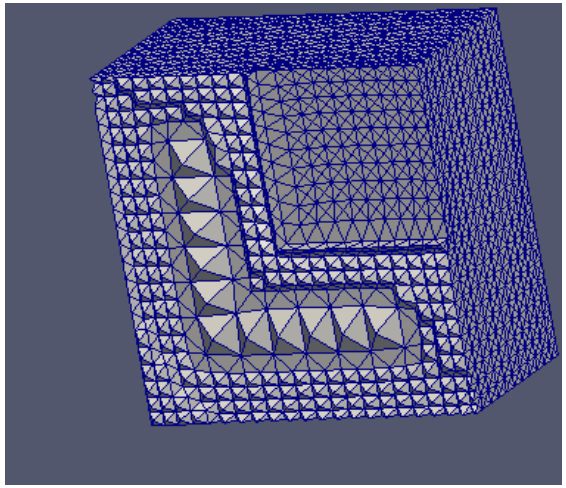




Gmsh

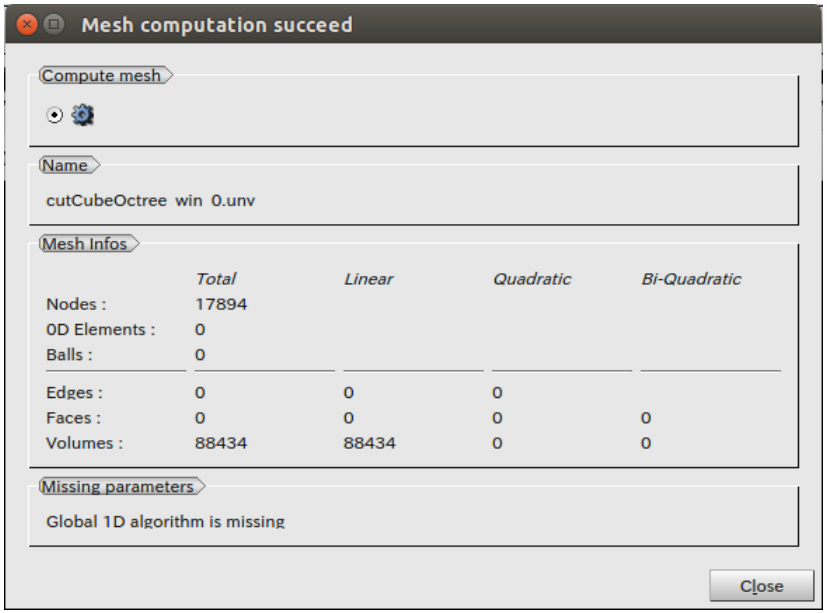


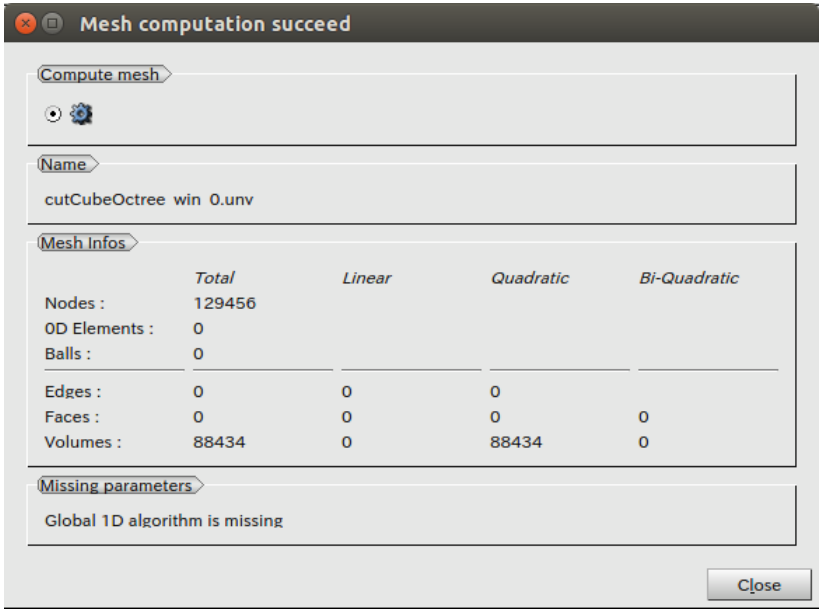
salome



cfMesh(tetMesh)

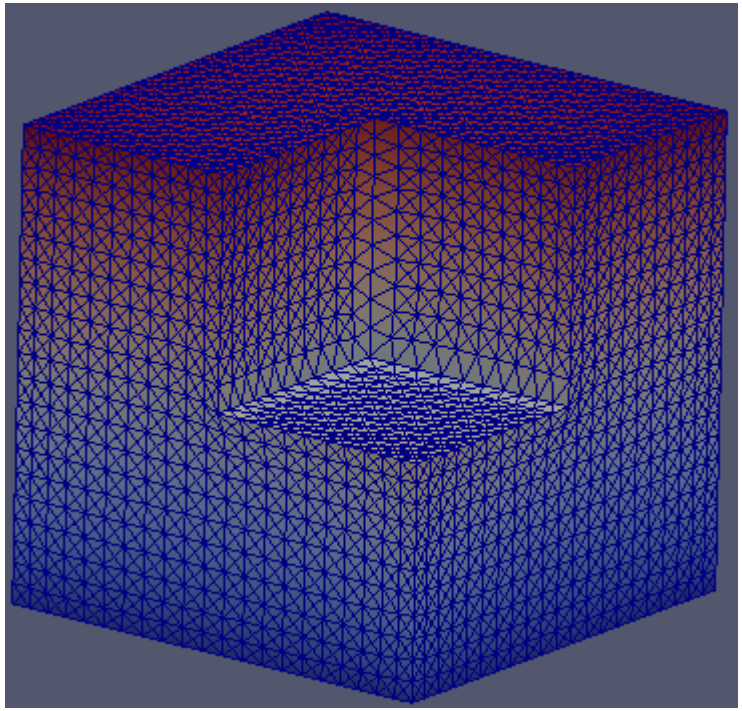
二次要素に変換

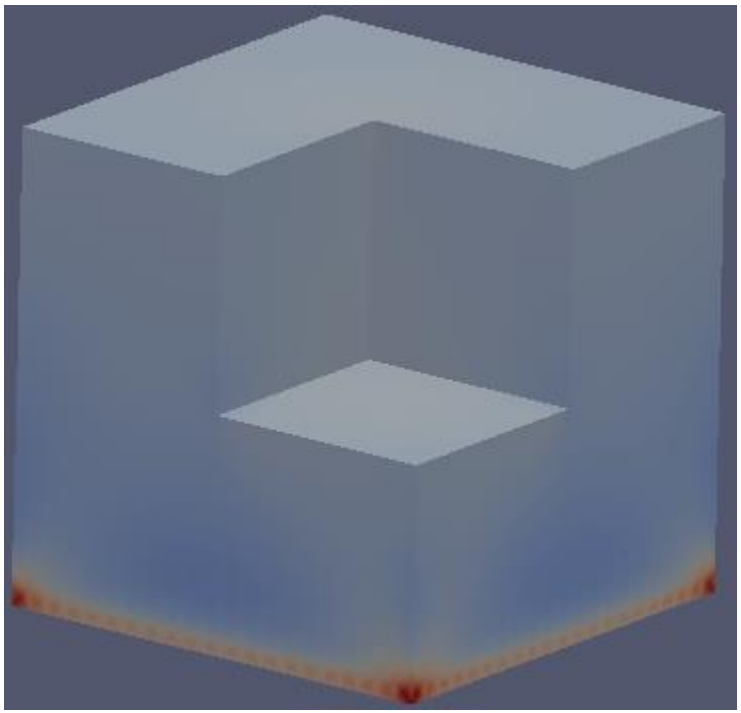

Mesh computation succeed
Compute mesh
Name
cutCubeOctree win 0.unv
Mesh Infos
Total Linear Quadratic Bi-Quadratic
Nodes : 17894
0D Elements : 0
Balls : 0
Edges : 0 0 0
Faces : 0 0 0 0
Volumes : 88434 88434 0 0
Missing parameters
Global 1D algorithm is missing
Close


Mesh computation succeed
Compute mesh
Name
cutCubeOctree win 0.unv
Mesh Infos
Total Linear Quadratic Bi-Quadratic
Nodes : 129456
0D Elements : 0
Balls : 0
Edges : 0 0 0
Faces : 0 0 0 0
Volumes : 88434 0 88434 0
Missing parameters
Global 1D algorithm is missing
Close

cfMesh(tetMesh)を使って構造解析用メッシュを作成する

- ロバストにメッシュ作成が可能
- Salomeで二次メッシュに変換すれば精度もある程度出るのではないか？





Patch名を修正する

⇒V1.1からはrenameBoundaryで設定可能

sedコマンドを利用

sed -i -e "s/empty/patch/g" mesh.fms

mesh.fms内の文字列“empty”を検索し”patch”に置き換える
-iをつけることでmesh.fmsを上書きする

emptyをpatchに変更するスクリプト例

emptyTopatch

```
#!/bin/sh

sed -i -e "s/empty/patch/g" $1
```

$1とすることで引数を受け取る

$./emptyTopatch mesh.fms

meshDictで設定できる項目

```
FoamFile
{
    version    2.0;
    format     ascii;
    class      dictionary;
    location   "system";
    object     meshDict;
}
```

固定記述

※最大セルサイズが基準となり、それ以降は指定した値より小さいサイズになる

```
surfaceFile         "mesh.fms";
maxCellSize 20.0;

boundaryCellSize 1.0;
boundaryCellSizeRefinementThickness 1.0;
minCellSize 10.0;
```

形状ファイル　stlまたはfms
最大セルサイズ（絶対値でサイズを指定する）

境界層セルサイズ（オプション）
境界層の厚さ（オプション）
最小セルサイズ（オプション）

```
boundaryLayers
{
}
```

境界層の設定

```
localRefinement
{
}
```

Patch名によるサイズ設定

```
objectRefinements
{
}
```

オブジェクトによるサイズ設定

```
anisotropicSources
{
}
```

異方性メッシュの設定

```
renameBoundary
{
}
```

Patch名とpatchタイプの変更

```
workflowControls
{
}
```

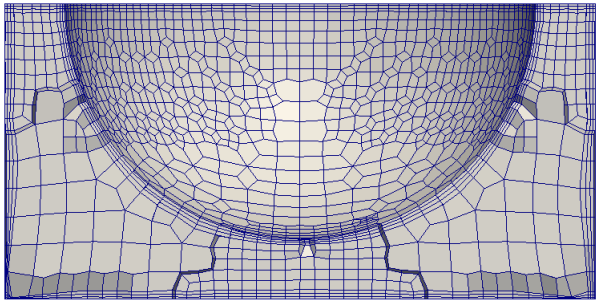
コントロール

boundaryLayersで設定できる項目

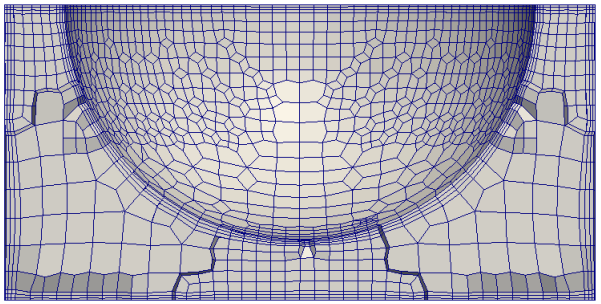
境界層の設定

```
boundaryLayers                                  すべての表面から境界層を作成する場合
{
maxFirstLayerThickness      0.5;                第1層の最大厚さ
nLayers      3;                                 層数
thicknessRatio      1.2;                        成長率

    patchBoundaryLayers                         各patchに境界層を作成する場合
    {
        patch1                                  境界層を生成するパッチ名
        {
            maxFirstLayerThickness      0.1;    第1層の最大厚さ
            nLayers      3;                     層数
            thicknessRatio      1.2;           成長率
            allowDiscontinuity  0;              不連続オプション　0or1　0:無効　1:有効
        }
    }

    optimiseLayer 1;                            境界層スムーズオプション　0:無効　1:有効
//    untangleLayers 1;                         デフォルトで有効のため無効化する？　0:無効　1:有効
    optimisationParameters
    {
        nSmoothNormals 5;                       法線方向スムーズ回数　デフォルト:5
        maxNumIterations 5;                     最大繰り返し回数　デフォルト:5
        featureSizeFactor 0.3;                  メッシュサイズと境界層サイズの比　0≦x＜1　デフォルト0.3
        reCalculateNormals 1;                   法線方向計算スイッチ　0or1　0:無効　1:有効　デフォルト1
        relThicknessTol 0.1;                    隣接する境界層厚さの最大差　0≦x＜1
    }
}
```

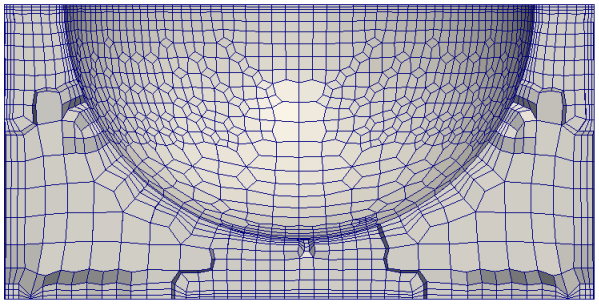
optimisationParameters



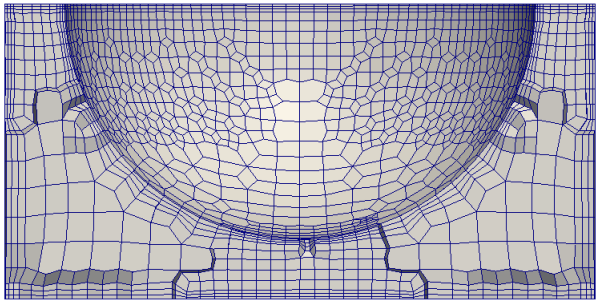
nSmoothNormals 5;
maxNumIterations 5;
featureSizeFactor 0.1;
reCalculateNormals 1;
relThicknessTol 0.1;



nSmoothNormals 5;
maxNumIterations 5;
featureSizeFactor 0.9;
reCalculateNormals 1;
relThicknessTol 0.1;

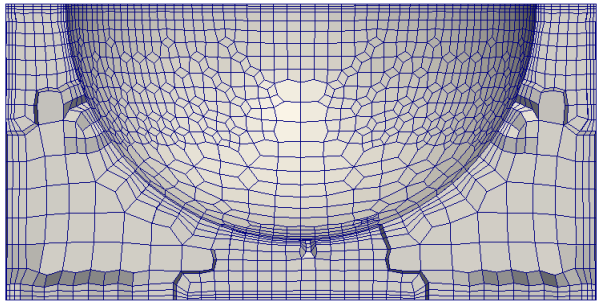


nSmoothNormals 5;
maxNumIterations 5;
featureSizeFactor 0.1;
reCalculateNormals 1;
relThicknessTol 0.9;

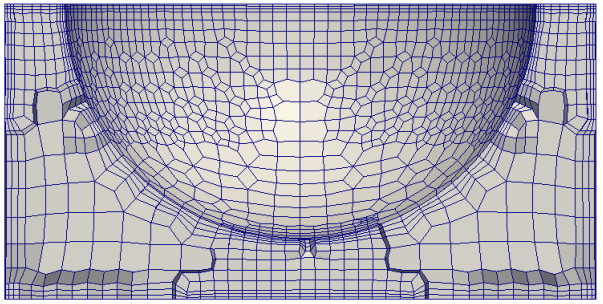


nSmoothNormals 5;
maxNumIterations 5;
featureSizeFactor 0.9;
reCalculateNormals 1;
relThicknessTol 0.9;

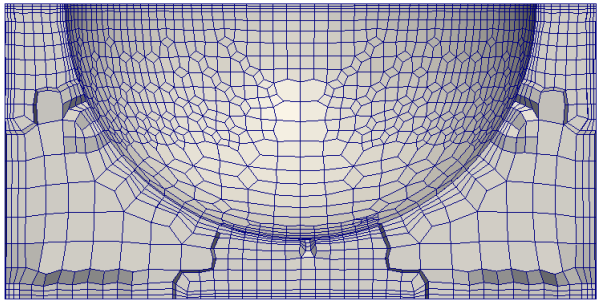
optimisationParameters



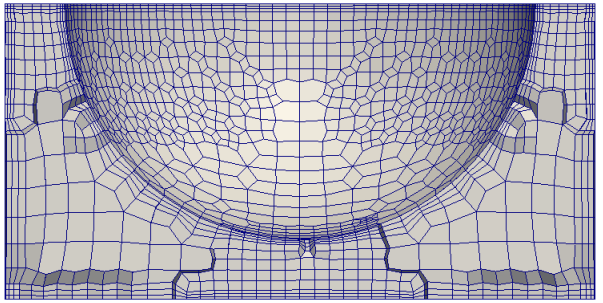
nSmoothNormals 5;
maxNumIterations 5;
featureSizeFactor 0.9;
reCalculateNormals 1;
relThicknessTol 0.9;



nSmoothNormals 5;
maxNumIterations 5;
featureSizeFactor 0.9;
reCalculateNormals 0;
relThicknessTol 0.9;

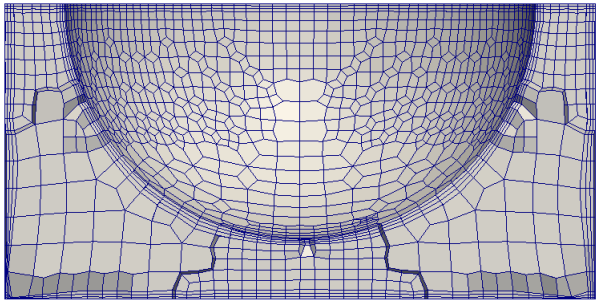


nSmoothNormals 1;
maxNumIterations 5;
featureSizeFactor 0.9;
reCalculateNormals 1;
relThicknessTol 0.9;

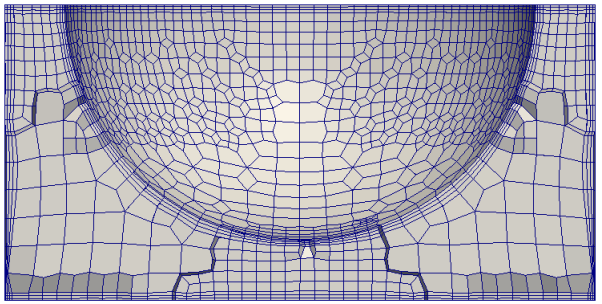


nSmoothNormals 5;
maxNumIterations 1;
featureSizeFactor 0.9;
reCalculateNormals 1;
relThicknessTol 0.9;

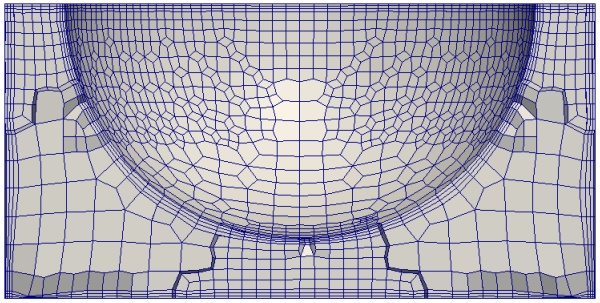
optimisationParameters



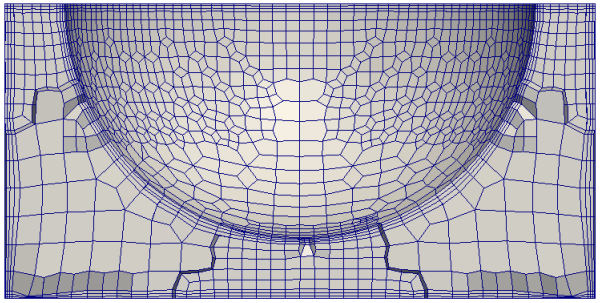
nSmoothNormals 5;
maxNumIterations 5;
featureSizeFactor 0.1;
reCalculateNormals 1;
relThicknessTol 0.1;



nSmoothNormals 5;
maxNumIterations 5;
featureSizeFactor 0.1;
reCalculateNormals 0;
relThicknessTol 0.1;



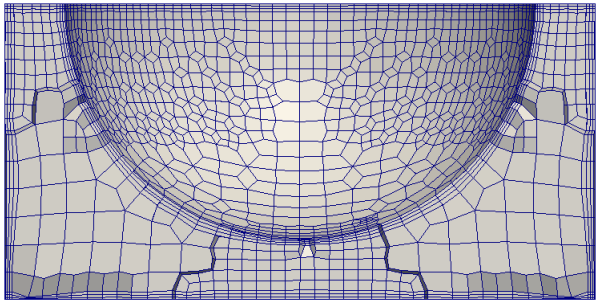
nSmoothNormals 1;
maxNumIterations 5;
featureSizeFactor 0.1;
reCalculateNormals 1;
relThicknessTol 0.1;



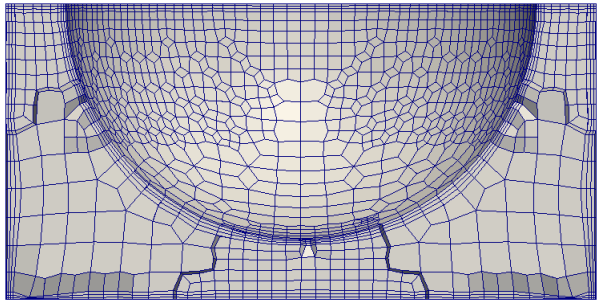
nSmoothNormals 5;
maxNumIterations 1;
featureSizeFactor 0.1;
reCalculateNormals 1;
relThicknessTol 0.1;

optimisationParameters

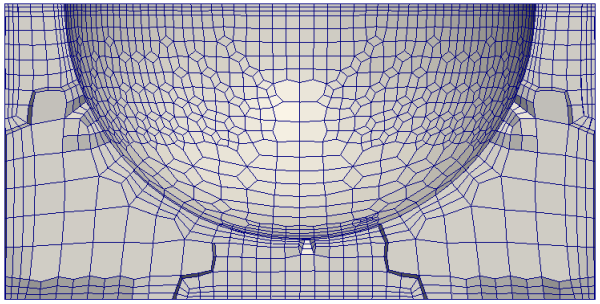
nSmoothNormals 5;
maxNumIterations 5;
featureSizeFactor 0.1;
reCalculateNormals 1;
relThicknessTol 0.1;



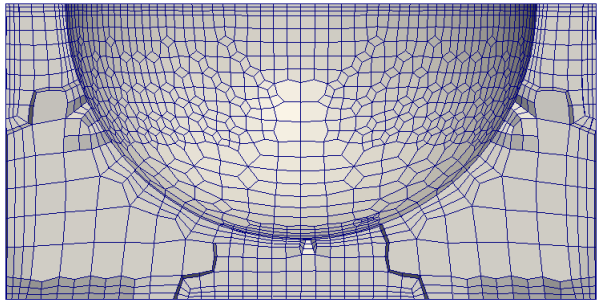
optimiseLayer 1;
untangleLayers 1;



optimiseLayer 1;
untangleLayers 0;



optimiseLayer 0;
untangleLayers 1;



optimiseLayer 0;
untangleLayers 0;

localRefinement で設定できる項目

```
localRefinement
{

    patch1
    {
        cellSize        0.25;
       additionalRefinementLevels        1;

       refinementThickness 4.5;
    }
 patch2
    {
        cellSize        0.125;
        additionalRefinementLevels        2;

        refinementThickness 4.5;
    }
}
```

各patchによるセルサイズの指定

セルサイズor細分化レベル

パッチから細分化する範囲（オプション）

objectRefinementsで設定できる項目

```
objectRefinements                オブジェクトによるサイズ指定
{
    coneExample
    {
        cellSize 3.75;           セルサイズ
        type cone;               円錐形状
        p0   (-100 1873 -320);   中心点
        radius0  200;            半径の長さ
        p1   (-560 1400 0);
        radius1   200;
    }
    boxExample
    {
        cellSize 3.75;           セルサイズ
        type box;                矩形形状
        centre  (500 500 150);   中心点
        lengthX  100;            各辺の長さ
        lengthY  150;
        lengthZ  200;
    }
    sphereExample
    {
        cellSize 3.75;           セルサイズ
        type sphere;             球形状
        centre  (0 700 0);       中心点
        radius  50;              半径の長さ
    }

    lineExample
    {
        cellSize 3.75;           セルサイズ
        type line;               ライン
        p0      (-750 1000 450); 始点
        p1      (-750 1500 450); 終点
        refinementThickness 4.5; 細分化する範囲（オプション）
    }

    hollowConeExample
    {
        additionalRefinementLevels 2;  細分化レベル
        type hollowCone;               穴あき円錐
        p0   (-100 1873 -320);         始点
        p1   (-560 1400 0);            終点
        radius0_Inner  200;            始点内径
        radius0_Outer  300;            始点外径
        radius1_Inner   200;           終点内径
        radius1_Outer   300;           終点外径
    }
}
```

他にもedgeMeshRefinementが利用可能

anisotropicSourcesで設定できる項目

```
anisotropicSources                  異方性メッシュの設定
{
    Box
    {
        type box;                   オブジェクトタイプ
        centre (2800 0 250);        中心座標
        lengthX 6000;               各辺の長さ
        lengthY 1000;
        lengthZ 200;
        scaleX 1;                   各辺のスケーリングファクター
        scaleY 1;
        scaleZ 0.3;
    }

    planeUpper
    {
        type plane;                 オブジェクトタイプ
        origin (0 0 250);           中心座標
        normal (0 0 1);             法線方向
        scalingDistance 125;        スケーリングする距離
        scalingFactor 0.5;          スケーリングファクター
    }
}
```

renameBoundaryで設定できる項目

```
renameBoundary
{
    defaultName     fixedWalls;
    defaultType     wall;

    newPatchNames
    {
        "inlet.*"
        {
            newName     inlet;
            newType     patch;
        }
        "outlet.*"
        {
            newName     outlet;
            newType     patch;
        }
    }
}
```

デフォルトのパッチ名
デフォルトのパッチタイプ

指定したパッチ名を新しいパッチ名、パッチタイプに変更

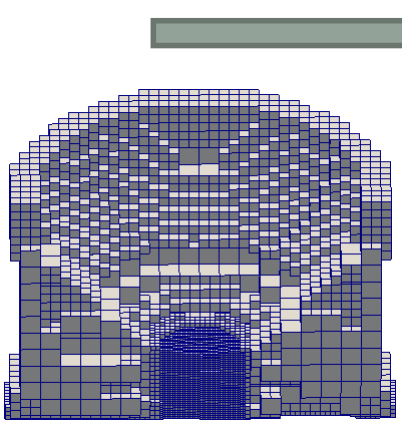
変更するパッチ名（ワイルドカード使用可能）

新しいパッチ名
新しいパッチタイプ

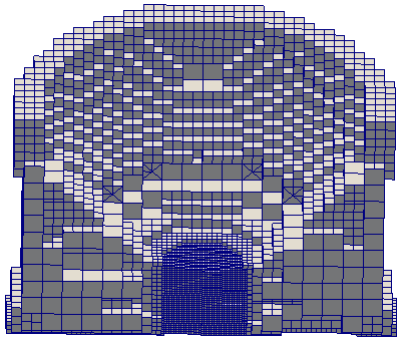
workflowControlsで設定できる項目

workflowControls	ワークフローコントロール
{	
//stopAfter templateGeneration;	八分木の作成と細分化
//stopAfter surfaceTopology;	
//stopAfter surfaceProjection;	サーフェスへ投影
//stopAfter patchAssignment;	境界パッチの割り当て
//stopAfter edgeExtraction;	エッジの抽出
//stopAfter boundaryLayerGeneration;	境界層の作成（1層のみ）
//stopAfter meshOptimisation;	メッシュの最適化（異方性メッシュ作成）
//stopAfter boundaryLayerRefinement;	境界層の細分化
//restartFromLatestStep 1;	メッシュ作成のリスタート　1：有効
}	

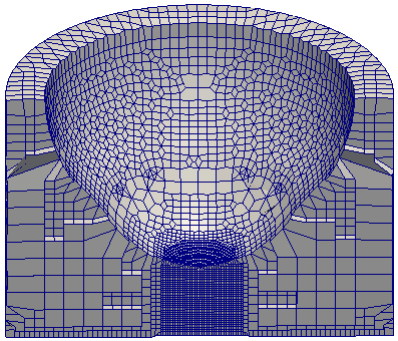
workflowControls



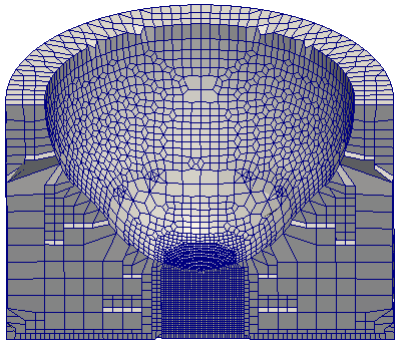
stopAfter
templateGeneration



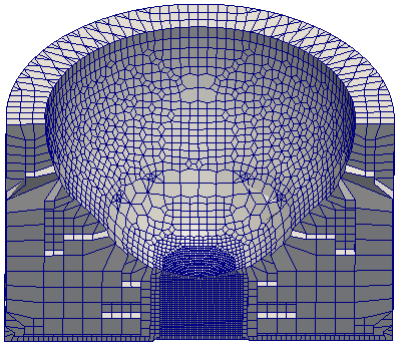
stopAfter
surfaceTopology



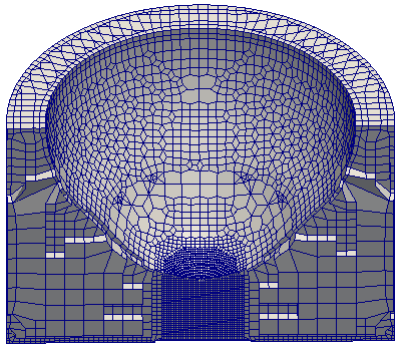
stopAfter
surfaceProjection



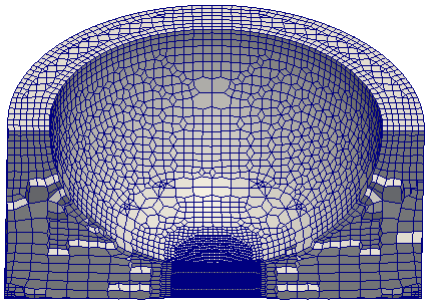
stopAfter
patchAssignment



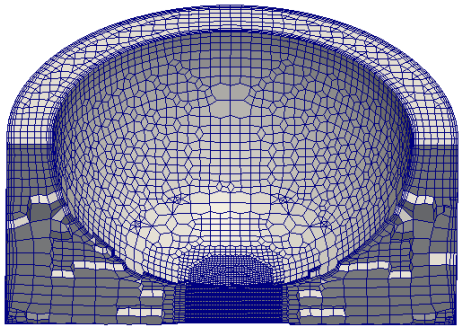
stopAfter
edgeExtraction



stopAfter
boundaryLayerGeneration



stopAfter
meshOptimisation



stopAfter
boundaryLayerRefinement