
OpenFOAMマイクロ講習： snappyHexMesh超入門 実習

2013年3月23日

オープンCAE勉強会@富山

中川慎二

注意

- この資料と例題ケースは、OpenFOAM 2.1.1 をベースにして作成しました。(2.0.0用および2.2.0用のケースもあります。)
- 最新版 OpenFOAM 2.2.0 (6th March 2013) では、snappyHexMeshが改良されていると言われています。
 - Many new features have been added to snappyHexMesh, particularly to improve feature snapping, layer insertion and baffle creation. Further details...

注意

- blockMeshDict 内で、設定の変更を容易にするため、変数を使っている。これを使うためには、システムのOpenFOAM設定で許可を与える必要がある。
- blockMesh の実行時に下記エラーメッセージが出た場合、ファイルの修正を行う。(root権限が必要)

--> FOAM FATAL IO ERROR:

Loading a shared library using case-supplied code is not enabled by default because of security issues. If you trust the code you can enable this facility by adding to the InfoSwitches setting in the system controlDict:

```
allowSystemOperations 1
```

The system controlDict is either

```
~/OpenFOAM/$WM_PROJECT_VERSION/controlDict
```

or

```
$WM_PROJECT_DIR/etc/controlDict
```

配布USBメモリの内容

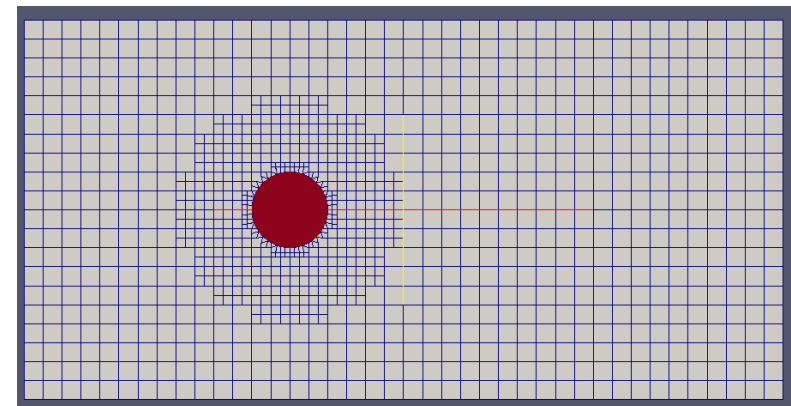
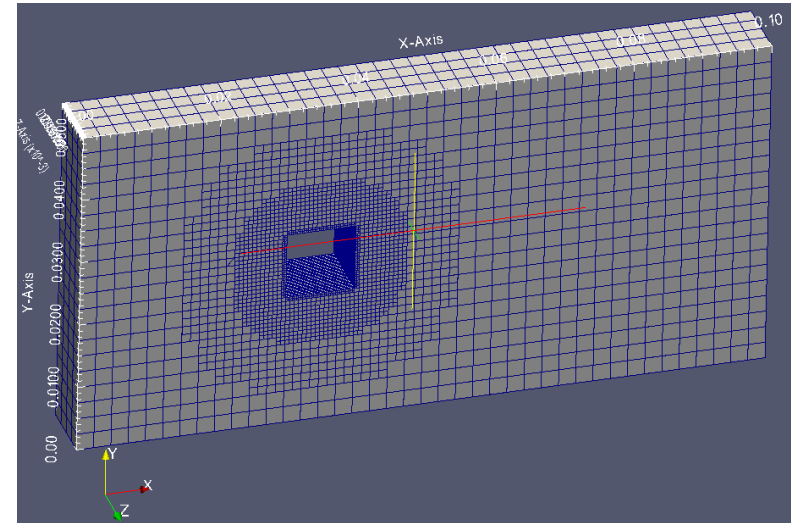
- Virtual_Machines_VM
 - Vmware仮想マシンを格納(2バージョン)
 - Ubuntu12.10 + OF 2.1.0
 - Ubuntu12.10 + OF 2.2.0
- OF200_case
 - OpenFOAM 2.0.0 用のケースディレクトリ
- OF210_case
 - OpenFOAM 2.1.0 用のケースディレクトリ
- OF220_case
 - OpenFOAM 2.2.0 用のケースディレクトリ

配布USBメモリの内容

- OpenFOAMが起動できる PC の使う場合
 - インストールされたOFのバージョンに応じたケースファイルを下記から選び、PCにコピーする
 - OF200_case
 - OF210_case
 - OF220_case
- VMware Playerを起動できるPCを使う場合
 - VMware player を起動し、使いたいバージョンの入った仮想マシンを開く。
 - Ubuntu12.10 + OF 2.1.0
 - Ubuntu12.10 + OF 2.2.0
 - 移動したか、コピーしたかと聞かれた場合には、コピーしたと答える(default)

例題

- snappyTestRect
 - 計算領域: 直方体
 - 内部に四角柱を設置
 - 四角柱の周りにセルを生成
- snappyTestCyl
 - 計算領域: 直方体
 - 内部に円柱を設置
 - 円柱の周りにセルを生成

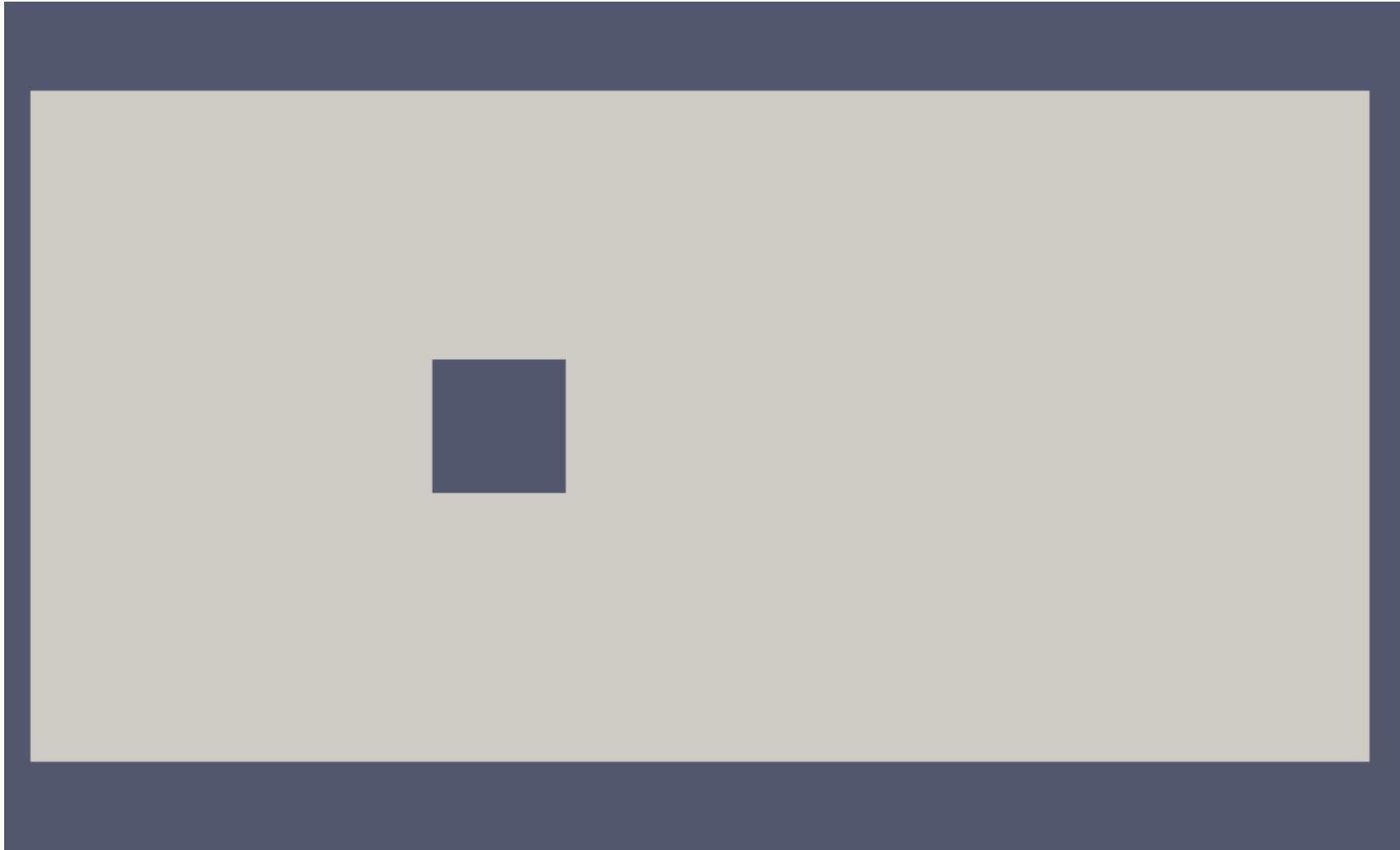


snappyTestRectケース:ファイル構成

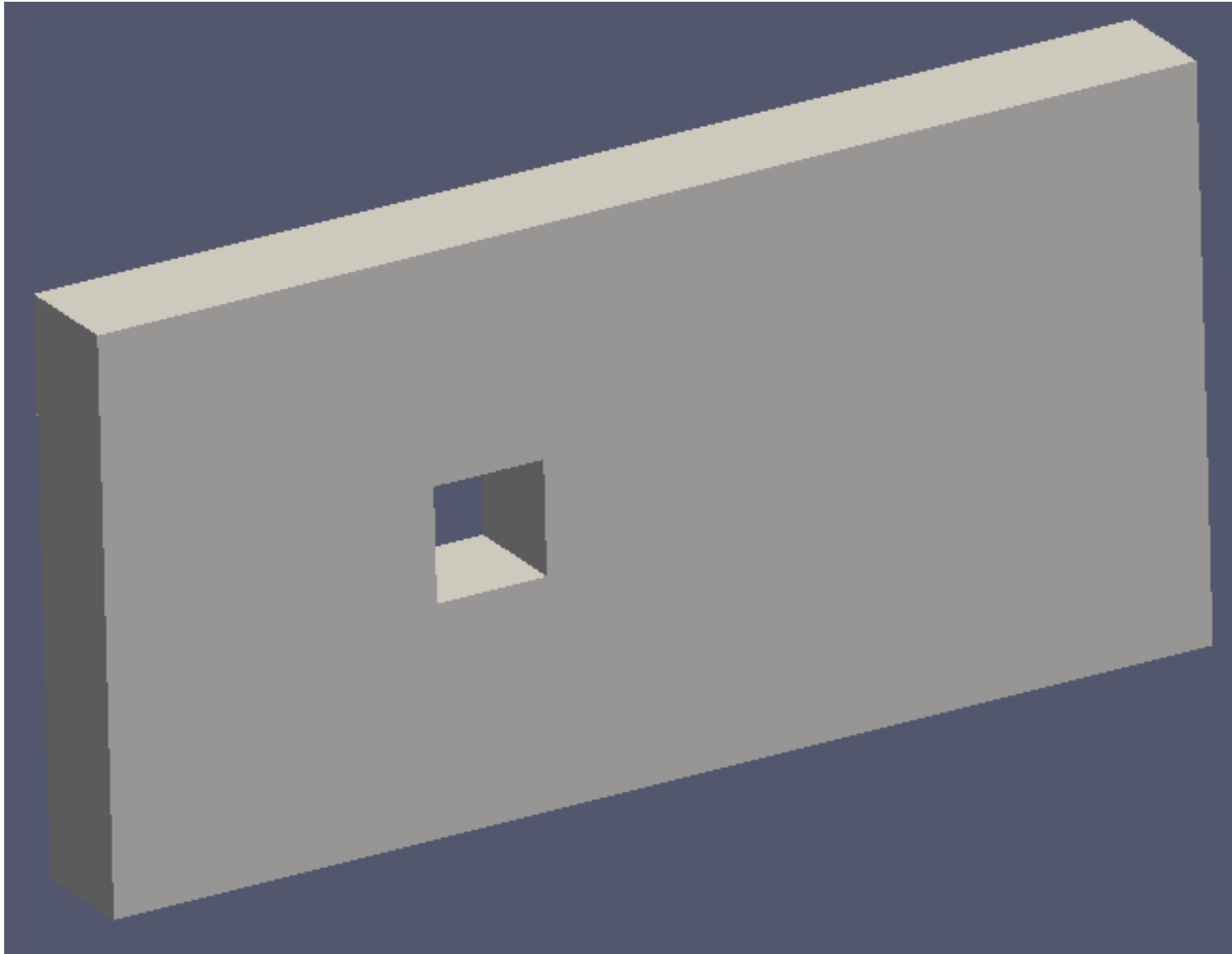
Name	Size	Type
0	2 items	folder
p	1.4 kB	C source code
U	1.5 kB	C source code
0.org	2 items	folder
p	1.4 kB	C source code
U	1.5 kB	C source code
constant	4 items	folder
polyMesh	2 items	folder
blockMeshDict	2.3 kB	C source code
boundary	1.8 kB	C source code
triSurface	1 item	folder
rectCylinder01.stl	2.5 kB	plain text document
RASProperties	945 bytes	C source code
transportProperties	917 bytes	C source code
system	5 items	folder
controlDict	1.2 kB	C source code
decomposeParDict	1.2 kB	C source code
fvSchemes	1.4 kB	C source code
fvSolution	1.3 kB	C source code
snappyHexMeshDict	11.3 kB	C source code
Allclean	311 bytes	shell script
snappyTraining	869 bytes	shell script

- 形状STLファイル
 - constant/triSurfaceに置く
 - rectCylinder01.stl
- 基準メッシュ設定
 - constant/polyMesh/blockMeshDict
- 詳細メッシュ設定
 - system/snappyHexMeshDict

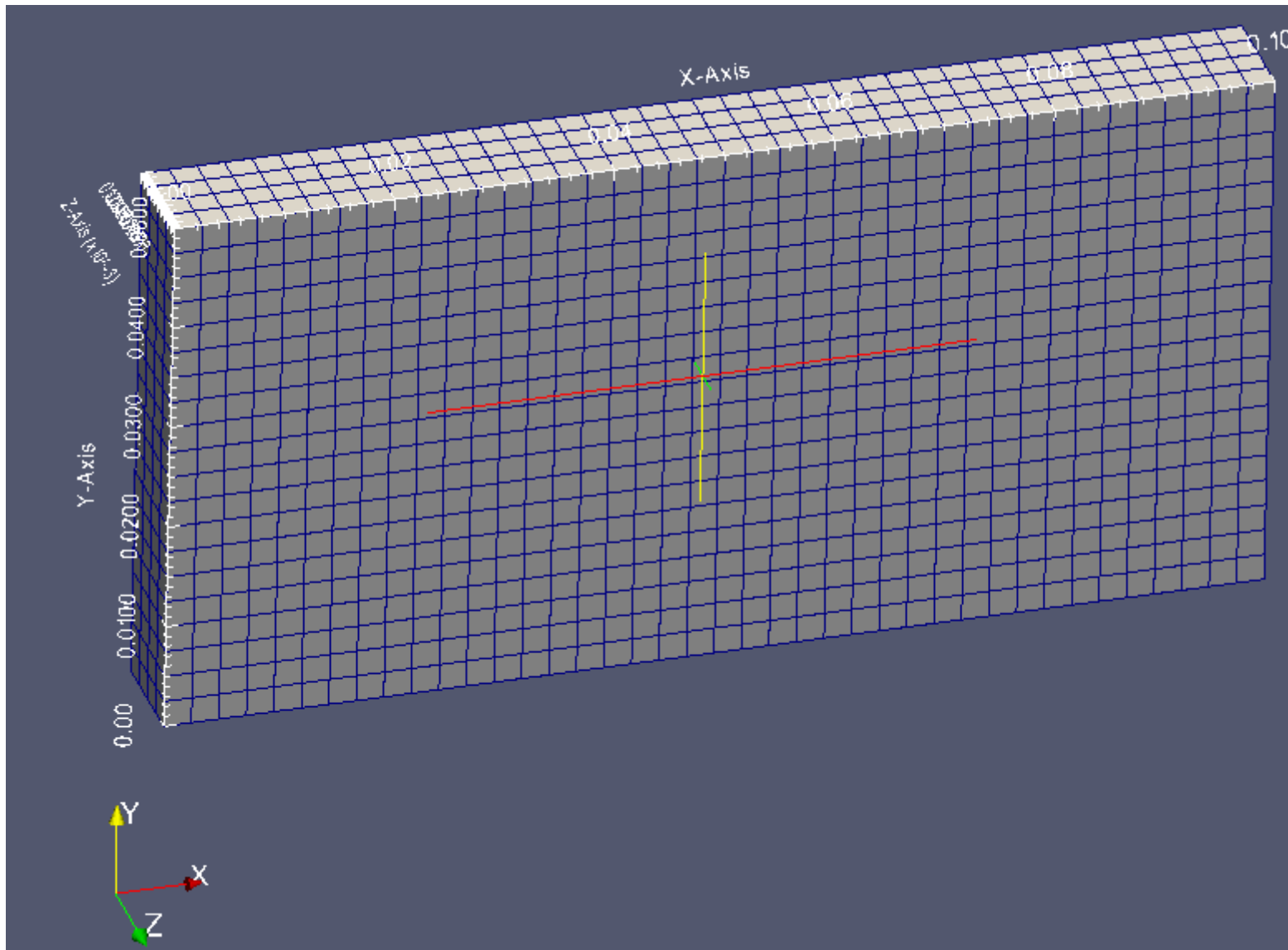
作成したい計算領域



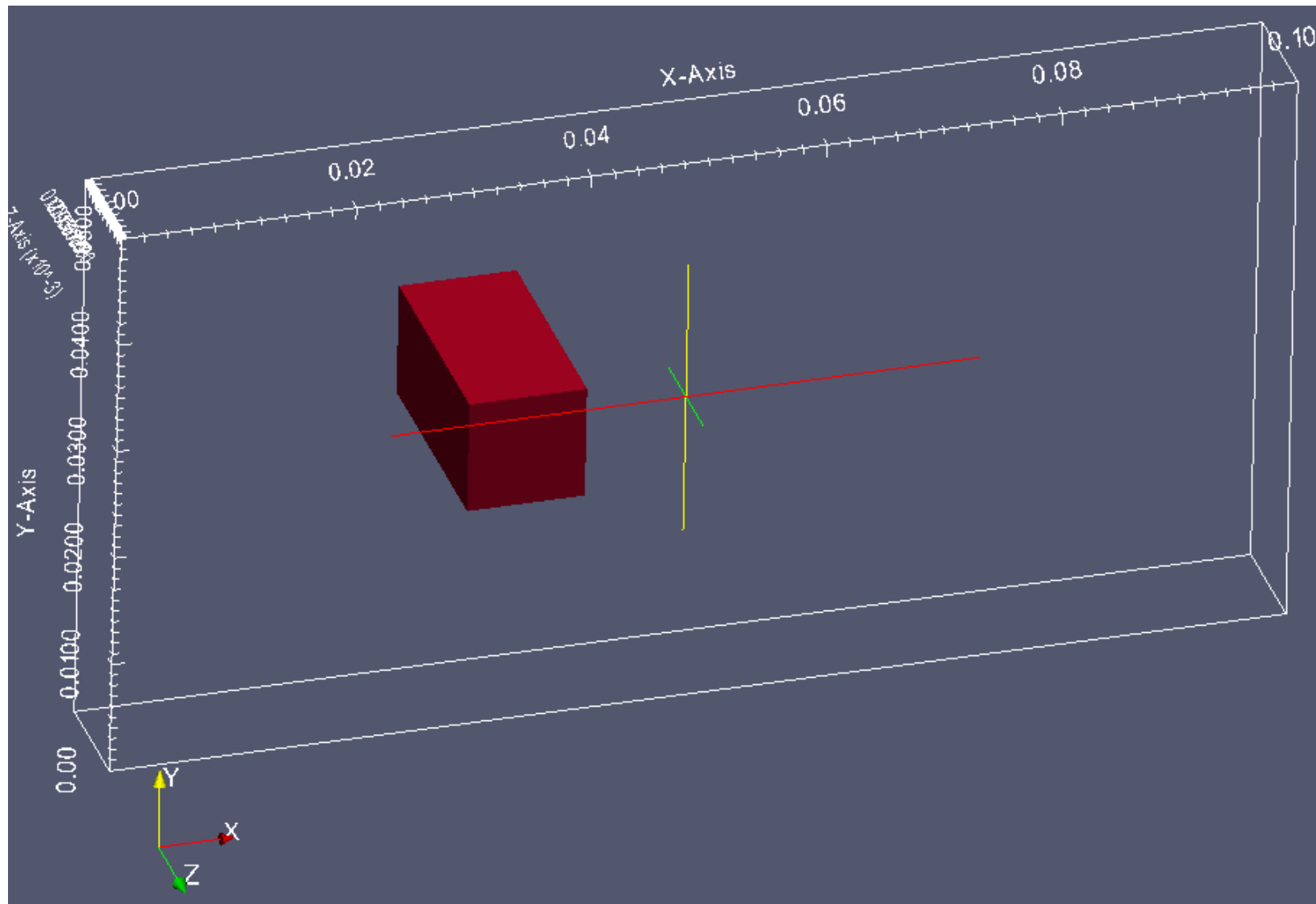
作成したい計算領域



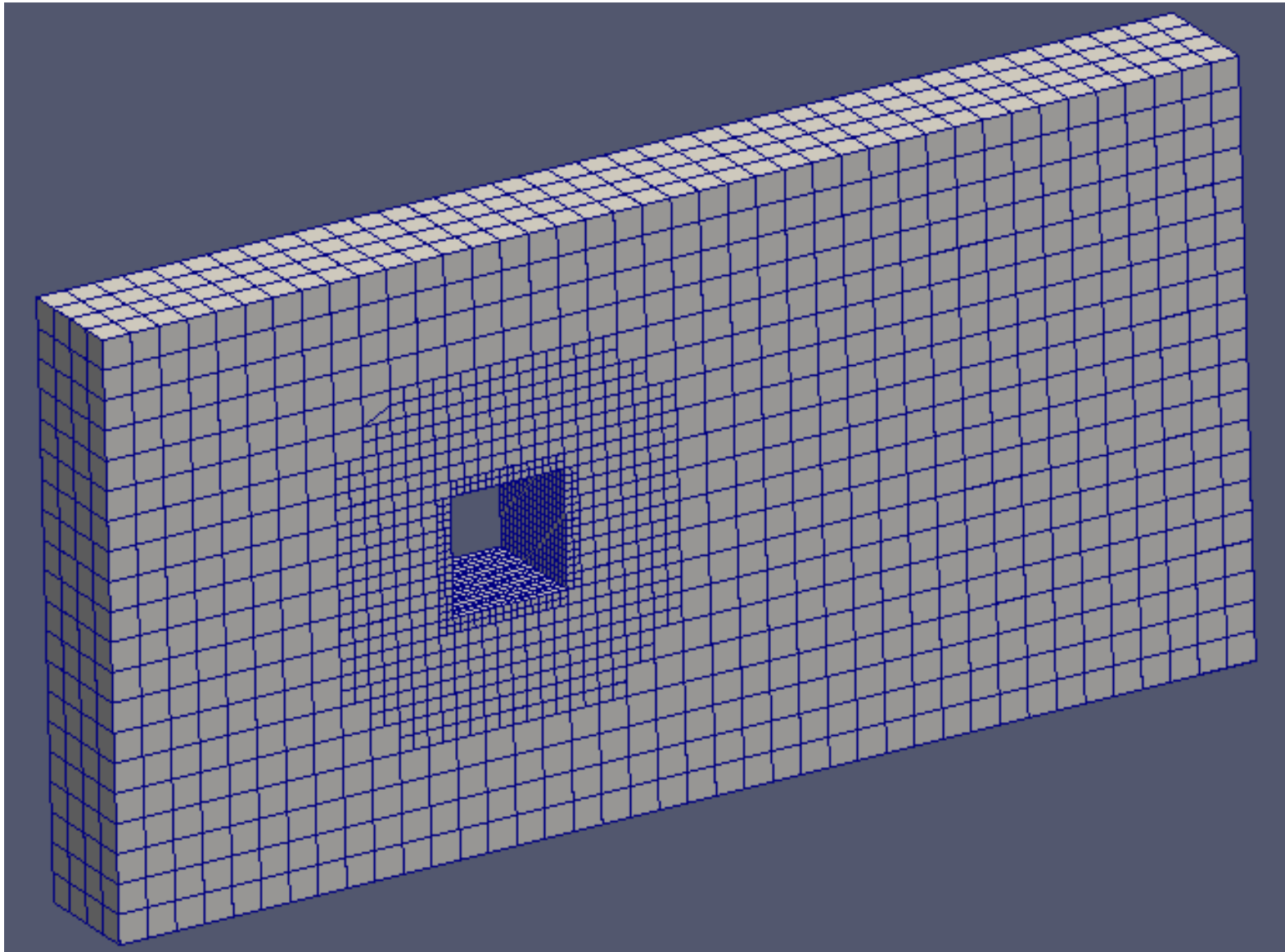
基準となるメッシュ (level 0)



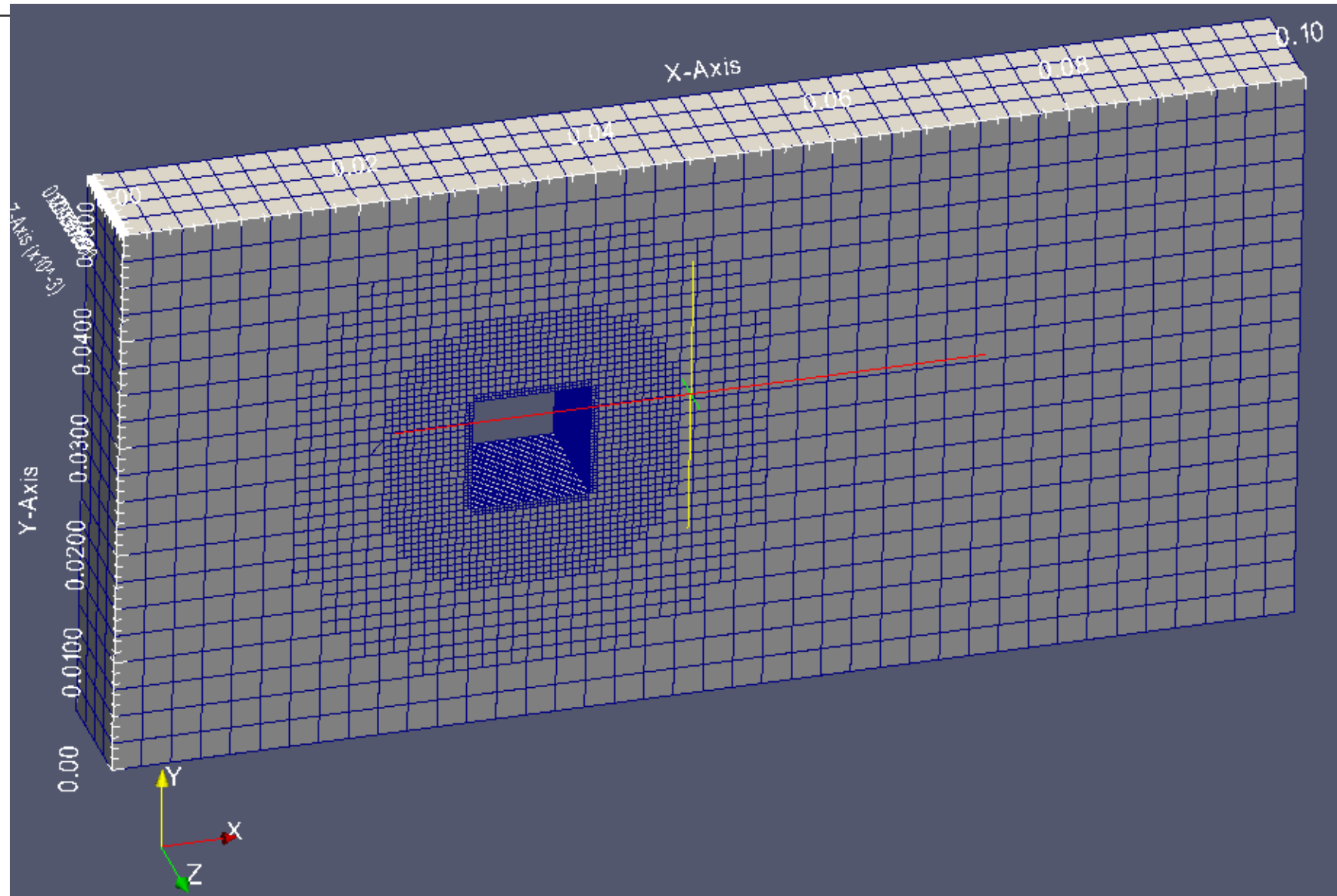
領域内に置いた角柱(STL)



完成図(例1)



完成図(例2)



実行手順

- 端末内で, ケースディレクトリsnappyTestRectに移動する
- ./snappyTraining と入力して, 実行する。自動的に下記コマンドが実行される。
 - blockMesh の実行
 - 基準メッシュ生成
 - surfaceFeatureExtract の実行
 - 角柱の特徴線抽出 → .eMeshファイル生成
 - snappyHexMesh の実行

スクリプト snappyTraining の内容

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

# Source tutorial run functions
. $WDM_PROJECT_DIR/bin/tools/RunFunctions

runApplication blockMesh
runApplication surfaceFeatureExtract -includedAngle 150 -writeObj
constant/triSurface/rectCylinder01.stl rectCylinder01
mv log.surfaceFeatureExtract surfaceFeatureExtract_rectCylinder01.log

runApplication snappyHexMesh
# runApplication snappyHexMesh -overwrite

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

#cp -rf 0.org 0

#runApplication `getApplication`
```

実行手順：詳細

- デスクトップのLX terminalをダブルクリックして、起動する。
- 例題ディレクトリ snappyTestRect に移動する。
cd snappyTestRect
- ./snappyTraining と入力して、実行する。自動的に下記コマンドが実行される。
 - blockMesh の実行
 - 基準メッシュ生成
 - surfaceFeatureExtract の実行
 - 角柱の特徴線抽出 → .eMeshファイル生成
 - snappyHexMesh の実行

作業

```
user@Lubuntu1210F210-VM: ~/snappyTestRect
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
OpenFOAM 2.1.0 enabled
user@Lubuntu1210F210-VM:~$ cd snappyTestRect
user@Lubuntu1210F210-VM:~/snappyTestRect$ ./snappyTraining
Running blockMesh on /home/user/snappyTestRect
Running surfaceFeatureExtract on /home/user/snappyTestRect
Running snappyHexMesh on /home/user/snappyTestRect
*****
mesh creation with snappyHexMesh is completed.

Please check log files!
Please check mesh with paraFoam

*****
user@Lubuntu1210F210-VM:~/snappyTestRect$ paraFoam
```

ディレクトリ移動

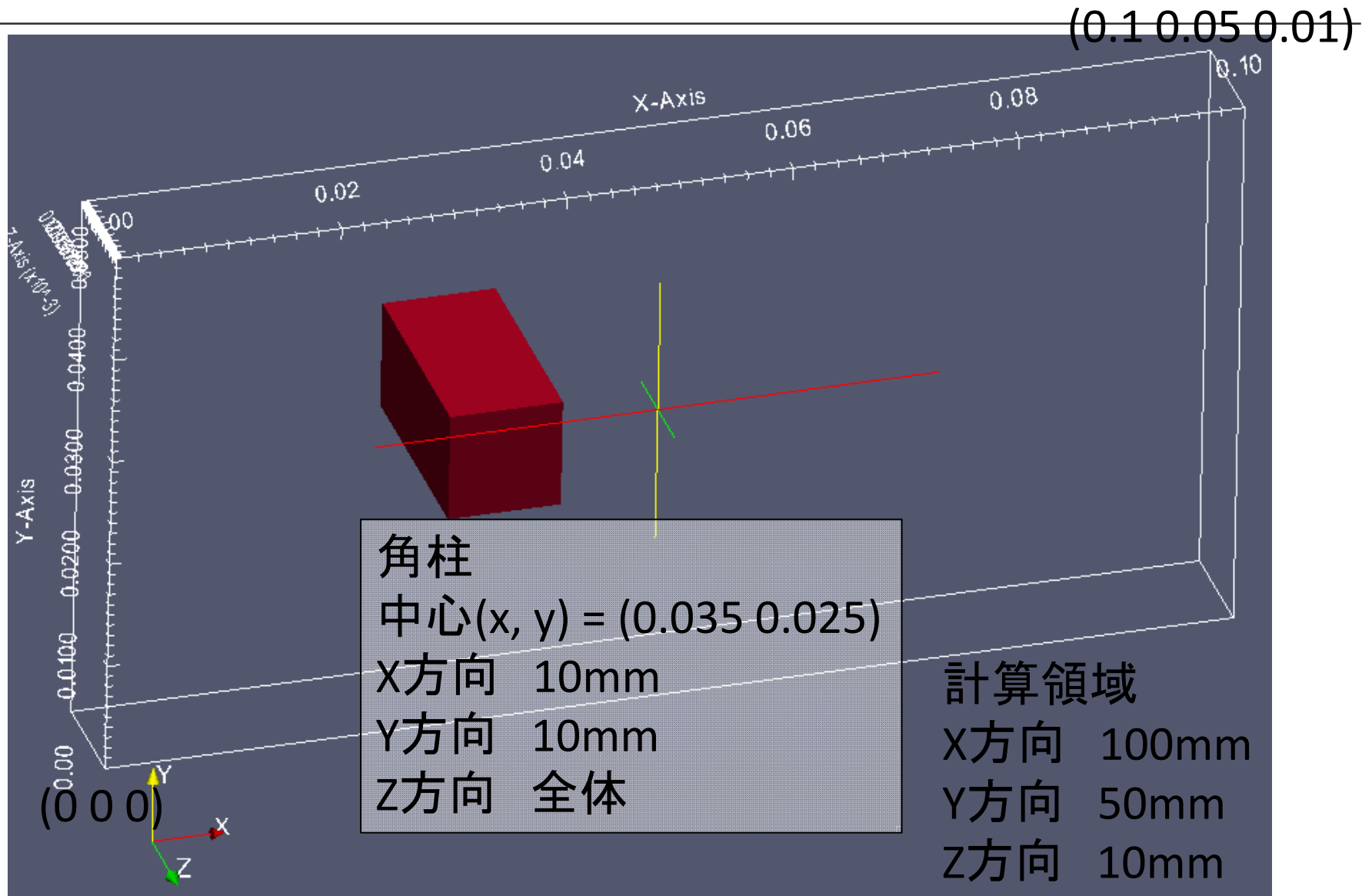
作業開始

可視化

実行手順(おまけ)

- 作業に失敗したときや, 条件を変えて再度実行したい場合には, ./Allclean と入力して, 実行する。
- 自動的に不要なファイルが削除され, 元の状態に戻る。

領域内に置いた角柱(STL)



全体の構造を考える

- 計算領域

- 直方体

- 座標範囲[m] $(0\ 0\ 0) - (0.1\ 0.05\ 0.01)$
 - 大きさ[m] $(x\ y\ z) = (0.1\ 0.05\ 0.01)$

- 基準セル

- 全体を立方体で埋める
 - 1辺の長さ 0.0025m ← 挿入する物体の1辺を4分割
 - 計算領域全体の分割数 $(40\ 20\ 4) \rightarrow 3200$

- 各面ごとを, 一つの境界面(patch)とする

blockMeshDict

```
convertToMeters 1;

//- number of mesh
nX 40; nY 20; nZ 4;

//- minimum coordinate [m]
_xMin 0.0; _yMin 0.0; _zMin 0.0;

//- calc domain length: base length [m]
base 0.010; // 10 mm

//- length of calc domain: coefficient
CLX 10.0;CLY 5.0;CLZ 1.0;

//- maximum coordinate [m]
_xMax #calc "$_xMin + $CLX*$base" ;
_yMax #calc "$_yMin + $CLY*$base" ;
_zMax #calc "$_zMin + $CLZ*$base" ;

vertices (
    ($_xMin $_yMin $_zMin) // 0
    ($_xMax $_yMin $_zMin) // 1
    ($_xMax $_yMax $_zMin) // 2
    ($_xMin $_yMax $_zMin) // 3

    ($_xMin $_yMin $_zMax) // 4
    ($_xMax $_yMin $_zMax) // 5
    ($_xMax $_yMax $_zMax) // 6
    ($_xMin $_yMax $_zMax) // 7
);

blocks(
    hex (0 1 2 3 4 5 6 7) ($nX $nY $nZ)
    simpleGrading (1 1 1)
);

Edges();
```

blockMeshDict

```
boundary(
  xMin
  {
    type patch;
    faces ( (0 4 7 3) );
  }
  xMax
  { type patch;
    faces ( (1 2 6 5) );
  }
  yMin
  {
    type wall;
    faces ( (0 1 5 4) );
  }
  yMax
  {
    type wall;
    faces ( (7 6 2 3) );
  }
  zMin
  {
    type cyclic;
    neighbourPatch zMax;
    faces ( (3 2 1 0) );
  }
  zMax
  {
    type cyclic;
    neighbourPatch zMin;
    faces ( (4 5 6 7) );
  }
);
```

全体の構造を考える

- 物体

- 直方体

- 入口(xMin)面から30mm下流に設定
 - 中心(x, y) = (0.035 0.025)
 - 10mm角の棒
 - 大きさ[m] (x y z) = (0.01 0.01 領域全体)
 - STLファイルを用意する
 - 今回は, 簡単な形状なので, シンプルなソフトで作成した
 - 無料の三次元CG/形状処理ソフトウェア StoneyDisigner (Winのみ)
 - <http://www.stoneydesigner.com/>

- セル

- 角柱の周囲は, 細かなセルを配置
 - 例: レベル1のセル 0.00125=1.25mm角, レベル2 0.625mm角
 - レベル2のセルを16個並べると, 角柱の1辺に相当する

snappyHexMeshの方針

- 角柱の周囲に詳細なメッシュ

snappyHexMeshDict よく使う項目

- geometry
 - STLファイルの読み込み
 - ファイル名, 内部での呼び方などを記載
- castellatedMeshControls
 - Features
 - STLファイルから作ったeMeshファイルを指定
 - 先にsurfaceFeatureExtractコマンドを実行しておく必要あり。
 - refinementSurfaces
 - 細分化したい面を指定
 - refinementRegions
 - 細分化したい領域を指定
 - distance 面からの距離が指定値以内
 - inside または outside 事前に用意した領域名で指定(領域内or外)
 - locationInMesh
 - メッシュを生成したい領域内部の点を指定
 - この指定間違いで, メッシュ生成できないことがある。

snappyHexMeshDict よく使う項目

- snapControls
 - nFeatureSnaplter
 - 特徴線を維持するには, 指定しておく必要がある
- addLayersControls
 - layers
 - レイヤーを追加したい面を指定

snappyHexMeshDict (geometry)

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

```
geometry  
{  
  rectCylinder01.stl //STL filename  
  {  
    type triSurfaceMesh;  
    name  cylinder;  
    regions  
    {  
      StoneyDesigner_solid  
      {  
        name  cylinder;  
      }  
    }  
  }  
};
```

snappyHexMeshDict (castellatedMeshControls)

```
castellatedMeshControls
{
    maxLocalCells 100000;
    maxGlobalCells 2000000;
    minRefinementCells 10;
    maxLoadUnbalance 0.10;
    nCellsBetweenLevels 1;

    features
    (
        {
            file "rectCylinder01.eMesh";
            level 2;
        }
    );

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

        resolveFeatureAngle 60;

        refinementRegions
        {
            cylinder
            {
                mode distance;
                levels ( (0.01 1) );
            }
        }

        locationInMesh (0.0001 0.0001 0.0001);
        allowFreeStandingZoneFaces true;
    }
}
```

snappyHexMeshDict(snap & addLayers)

```
snapControls
{
    nSmoothPatch 3;
    tolerance 4.0;
    nSolverIter 0;
    nRelaxIter 5;
    nFeatureSnapIter 10;
}
```

```
addLayersControls
{
    relativeSizes true;
    layers
    {
    }
}
```

```
expansionRatio 1.0;
finalLayerThickness 0.3;
minThickness 0.1;
nGrow 0;
featureAngle 30;
nRelaxIter 3;
nSmoothSurfaceNormals 1;
nSmoothNormals 3;
nSmoothThickness 10;
maxFaceThicknessRatio 0.5;
maxThicknessToMedialRatio 0.3;
minMedianAxisAngle 90;
nBufferCellsNoExtrude 0;
nLayerIter 50;
}
```

snappyHexMeshDict(meshQual. etc.)

```
meshQualityControls
{
    maxNonOrtho 65;
    maxBoundarySkewness 20;
    maxInternalSkewness 4;
    maxConcave 80;
    minVol 1e-13;
    minTetQuality 1e-30;
    minArea -1;
    minTwist 0.02;
    minDeterminant 0.001;
    minFaceWeight 0.02;

    minVolRatio 0.01;
    minTriangleTwist -1;

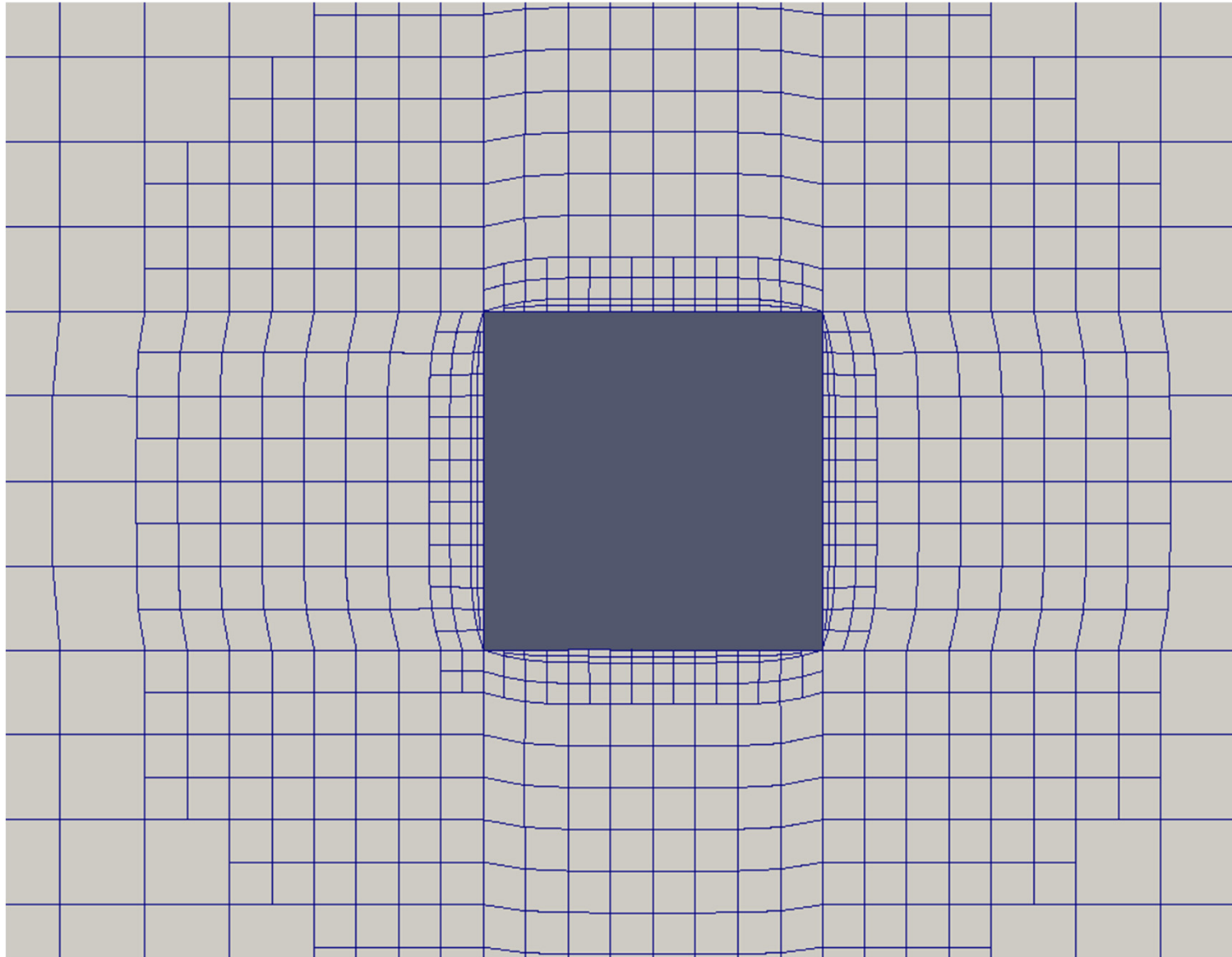
    nSmoothScale 4;
    errorReduction 0.75;
}

debug 0;
mergeTolerance 1e-6;
```

トライ

- snappyHexMeshDictの編集
 - たとえば
 - Levelを変えてみる
 - 領域再分割の範囲を変更してみる
 - 領域再分割の範囲指定方法を変えてみる
 - Layerを入れてみる
 - locationInMeshの座標を物体内部(0.0301 0.0201 0.0001)にしてみる
 - などなど
- ./Allclean の実行
- ./snappyTraining の実行
- paraFoam で変化の確認

Layer追加例



障害物の形状を変える：円柱

snappyTestCylケース:ファイル構成

Name	Size	Type	D
0	2 items	folder	TI
p	1.4 kB	C source code	TI
U	1.5 kB	C source code	TI
0.org	2 items	folder	TI
p	1.4 kB	C source code	TI
U	1.5 kB	C source code	TI
constant	4 items	folder	TI
polyMesh	2 items	folder	TI
blockMeshDict	2.3 kB	C source code	TI
boundary	1.8 kB	C source code	TI
triSurface	1 item	folder	TI
cylinder01.stl	62.2 kB	plain text document	TI
RASProperties	945 bytes	C source code	TI
transportProperties	917 bytes	C source code	TI
system	5 items	folder	TI
controlDict	1.2 kB	C source code	TI
decomposeParDict	1.2 kB	C source code	TI
fvSchemes	1.4 kB	C source code	SI
fvSolution	1.3 kB	C source code	SI
snappyHexMeshDict	11.3 kB	C source code	TI
Allclean	311 bytes	shell script	TI
snappyTraining	857 bytes	shell script	TI

• 形状STLファイル

– constant/triSurfaceに置く

– cylinder01.stl

• このファイルだけが異なる

• 基準メッシュ設定

– constant/polyMesh/blockMeshDict

• 詳細メッシュ設定

– system/snappyHexMeshDict

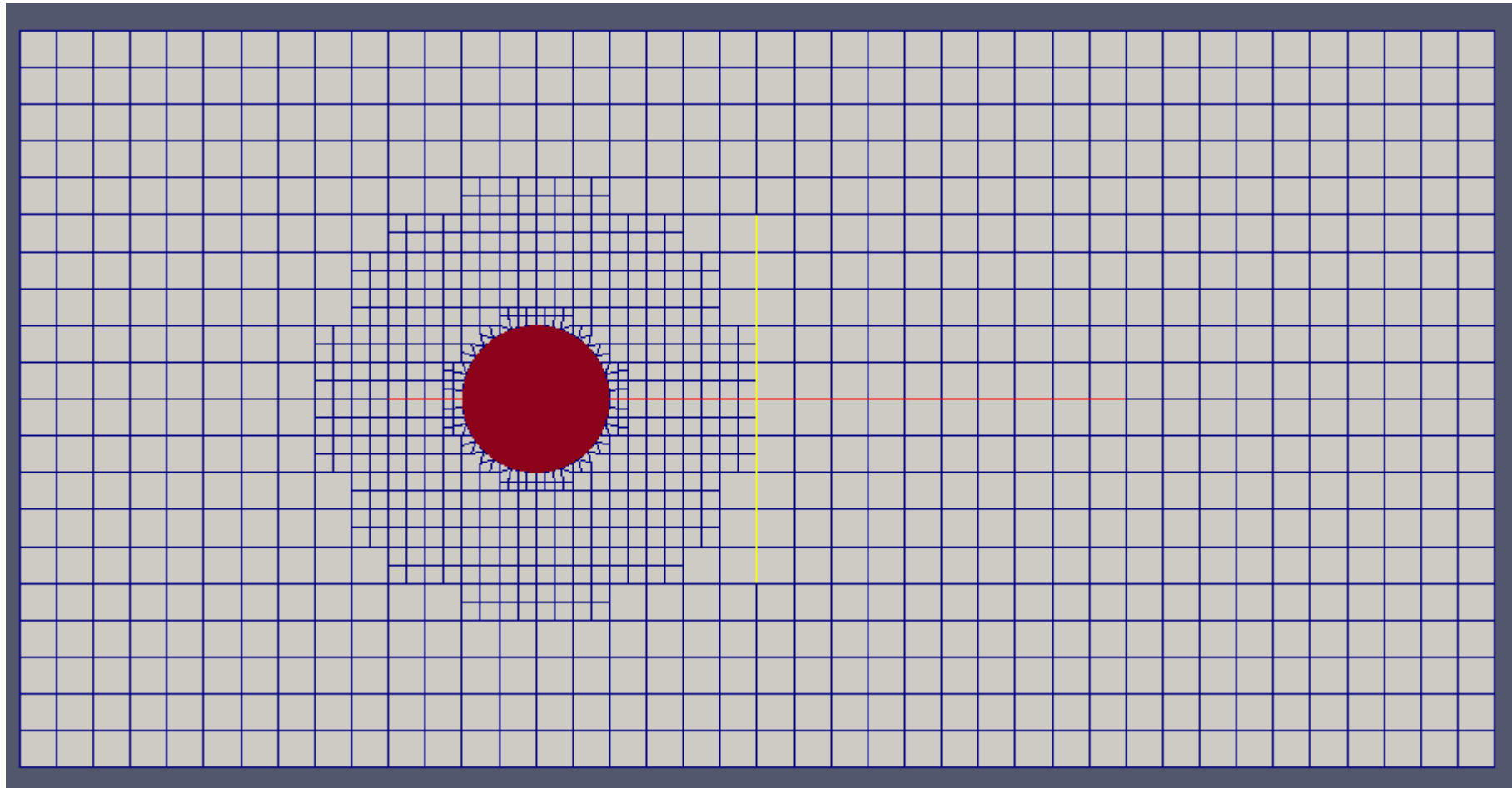
snappyTrainingスクリプトでも, STLファイル名が記載されているので, その名前を修正する。

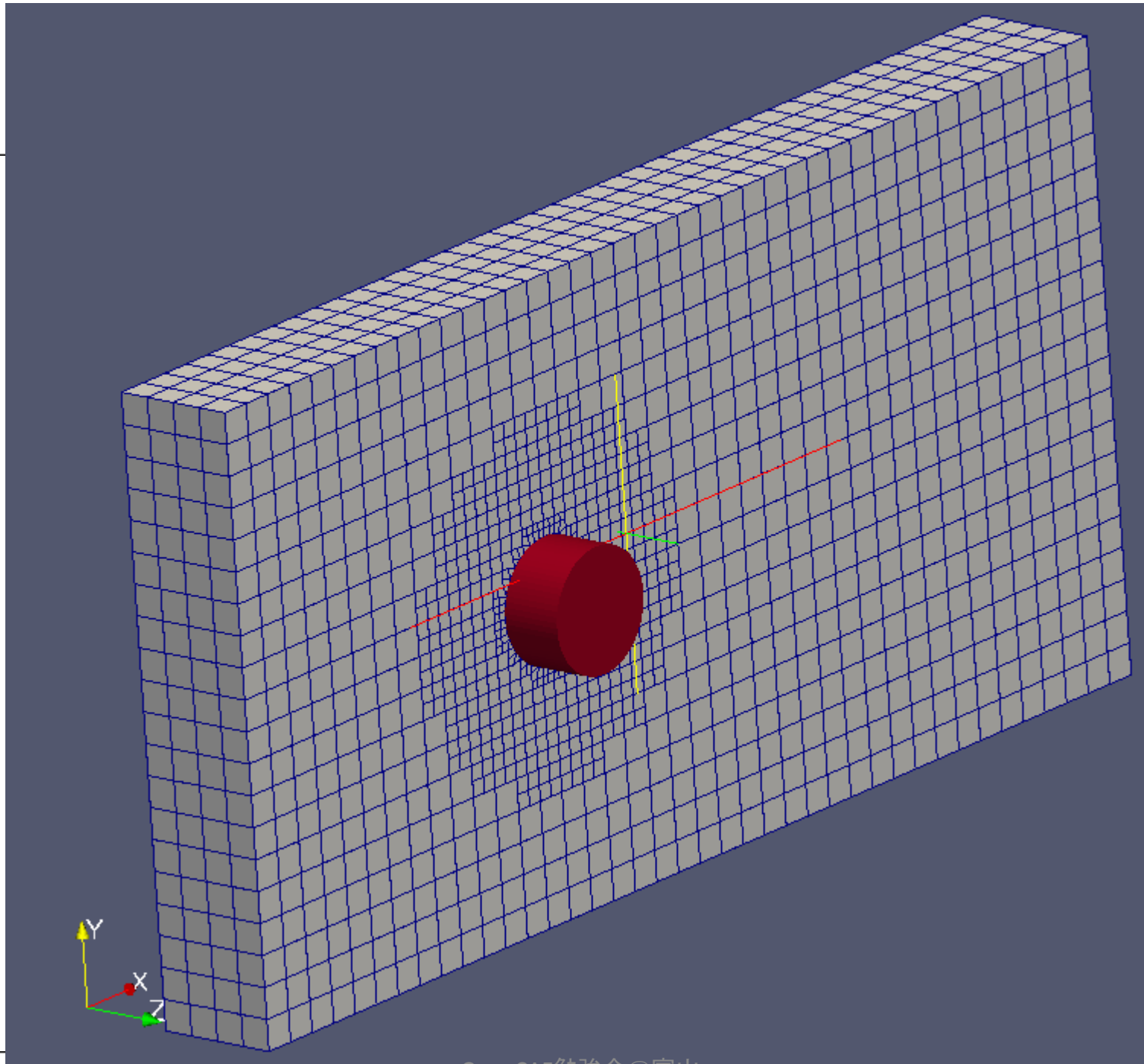
実行手順：詳細

- デスクトップのLX terminalをダブルクリックして、起動する。
- 例題ディレクトリ snappyTestCyl に移動する。
cd snappyTestCyl
- ./snappyTraining と入力して、実行する。自動的に下記コマンドが実行される。
 - blockMesh の実行
 - 基準メッシュ生成
 - surfaceFeatureExtract の実行
 - 角柱の特徴線抽出 → .eMeshファイル生成
 - snappyHexMesh の実行

作業

```
user@Lubuntu12100F210-VM:~/snappyTestRect$  
user@Lubuntu12100F210-VM:~/snappyTestRect$ cd .. ← ディレクトリ移動  
user@Lubuntu12100F210-VM:~$ cd snappyTestCyl ← ディレクトリ移動  
user@Lubuntu12100F210-VM:~/snappyTestCyl$ ./snappyTraining ← 作業開始  
Running blockMesh on /home/user/snappyTestCyl  
Running surfaceFeatureExtract on /home/user/snappyTestCyl  
Running snappyHexMesh on /home/user/snappyTestCyl  
*****  
mesh creation with snappyHexMesh is completed.  
  
Please check log files!  
Please check mesh with paraFoam  
  
*****  
user@Lubuntu12100F210-VM:~/snappyTestCyl$ paraFoam ← 可視化
```





メッシュをキレイに見るために

