ただで始める流体解析

snappyMultiRegionHeater チュートリアルの実施

本日の流れ

0. はじめに
1. チュートリアルを調査
2. コマンドによるモデル作成と設定
3. コマンドによる計算実施
4. まとめ，その他

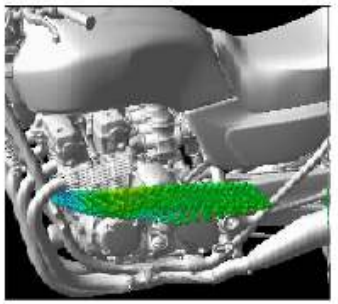
0. はじめに

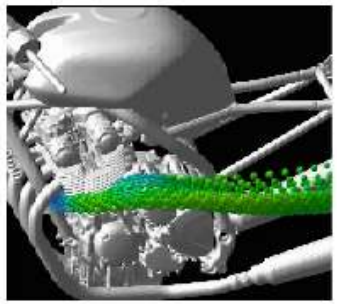
0：はじめに

伝熱，伝導のある流れ問題をやってみたいので，チュートリアルの“chtMultiRegionFoamのsnappyMultiRegionHeater”をやってみました。

将来的には，右にような計算をOpenFOAMで出来るようになれるといいですが。

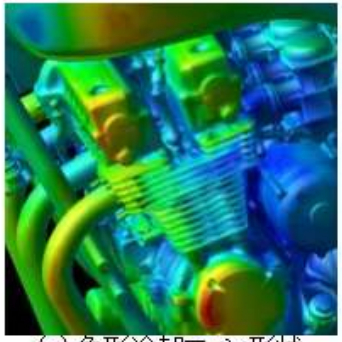
さていつになるでしょうか？





(a) 角形冷却フィン形状　(b) 丸形冷却フィン形状

Fig.2 冷却フィン周りの流れ（パーティクルトレース法）



(a) 角形冷却フィン形状　(b) 丸形冷却フィン形状

Fig.3 冷却フィン周りの熱伝達率分布

http://www.jsme.or.jp/ted/NL60/TED-Plaza_takahashi.htm

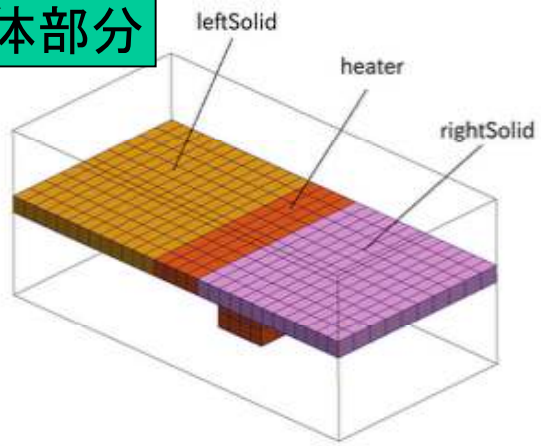
1．チュートリアルを調査

●まずはどんな問題か概要を見る

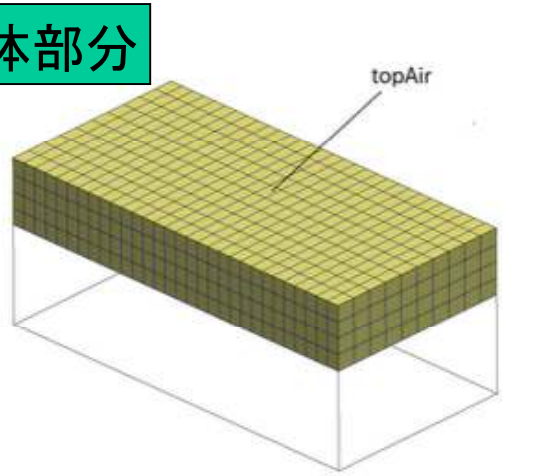
OpenFOAM Tutorial
ドキュメント作成プロジェクト

- 層流の非定常流体-固体熱連成解析
- メッシュ作成はsHMを利用
- 固体部分は鋼材，流体部分は空気
- topAirの部分に入口，出口，それ以外は壁
- 熱源設定はheater下部に設定

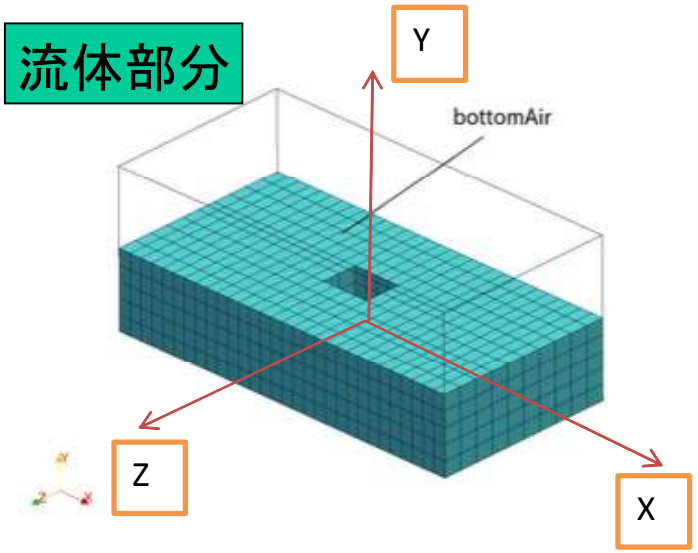
固体部分



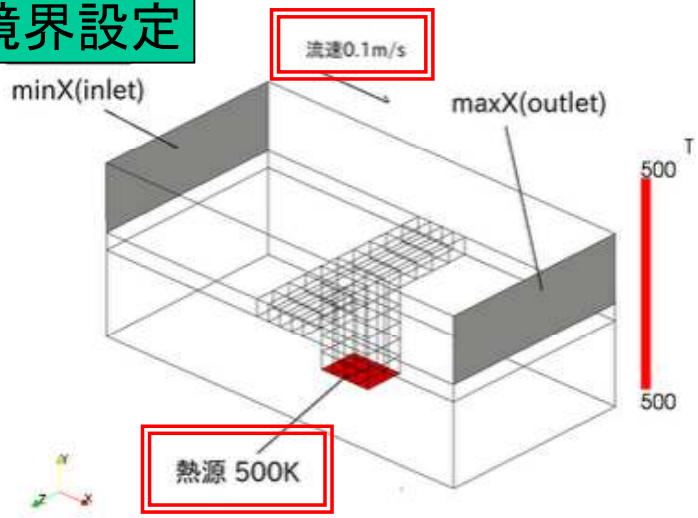
流体部分

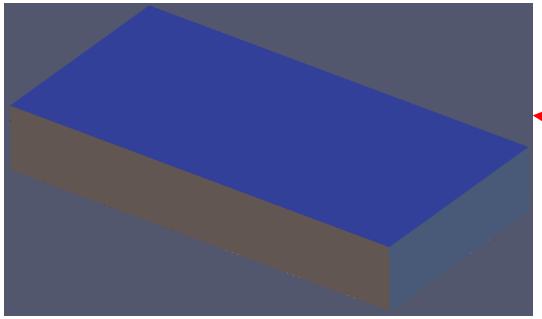


流体部分



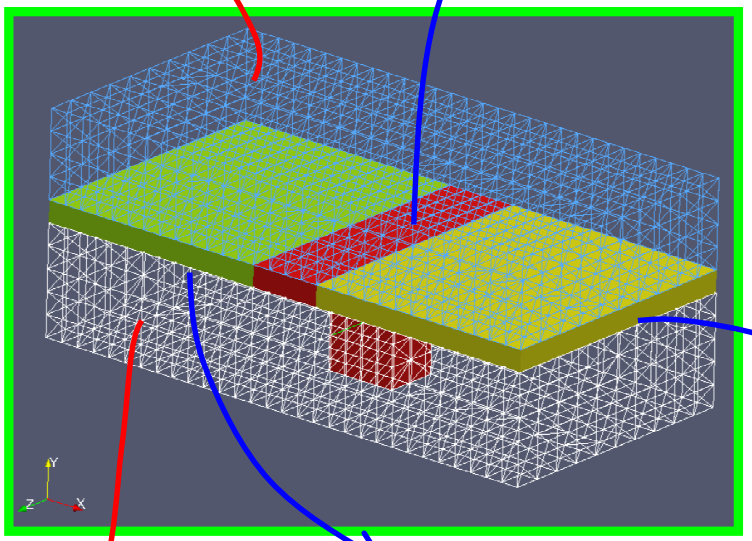
境界設定

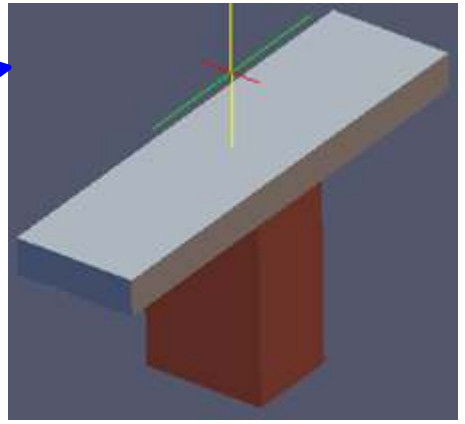




topAir

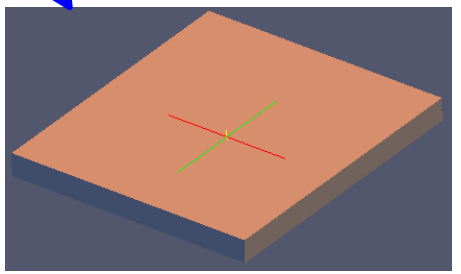
流体領域



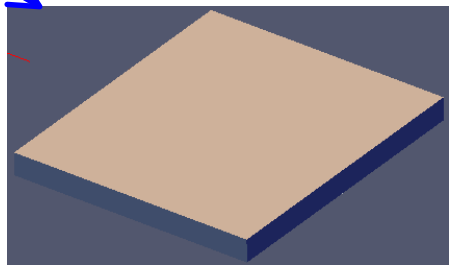


heater

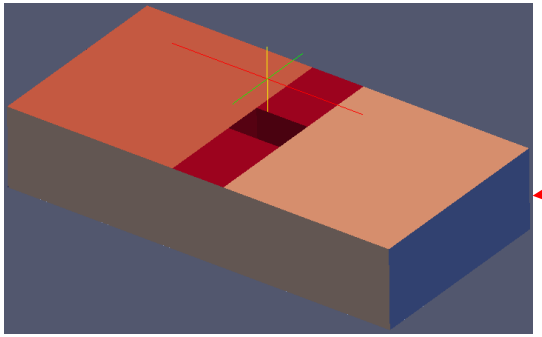
固体領域



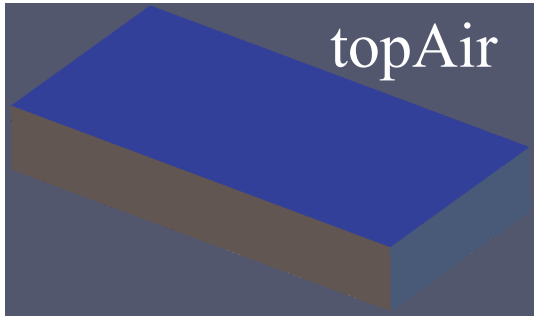
rightSolid



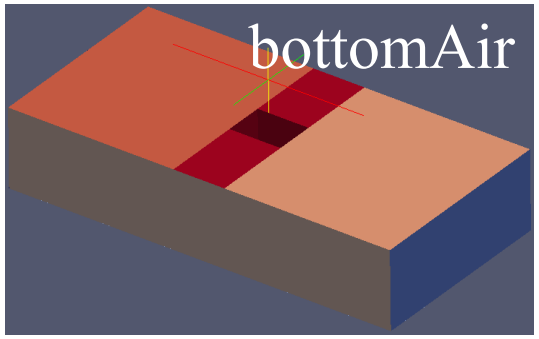
leftSolid



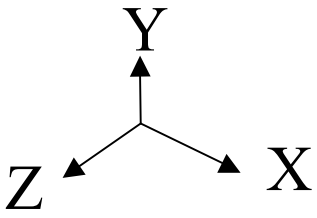
bottomAir

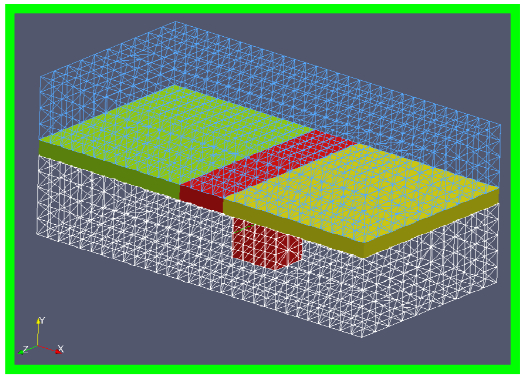


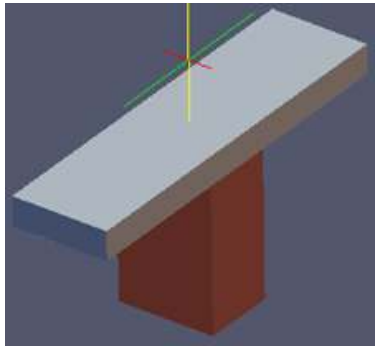
Bounds
X range: -0.1 to 0.1 (delta: 0.2)
Y range: 0.008 to 0.04 (delta: 0.032)
Z range: -0.05 to 0.05 (delta: 0.1)



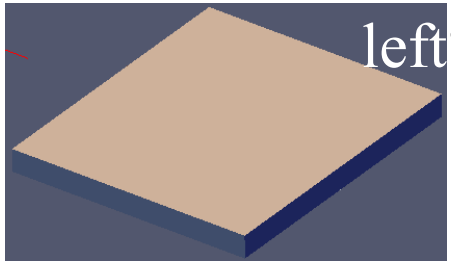
Bounds
X range: -0.1 to 0.1 (delta: 0.2)
Y range: -0.04 to 1.16e-18 (delta: 0.04)
Z range: -0.05 to 0.05 (delta: 0.1)



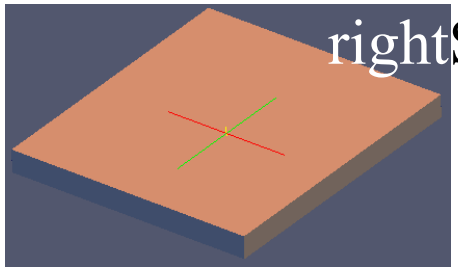




Bounds
X range: -0.0133 to 0.0133 (delta: 0.0267)
Y range: -0.04 to 0.008 (delta: 0.048)
Z range: -0.05 to 0.05 (delta: 0.1)

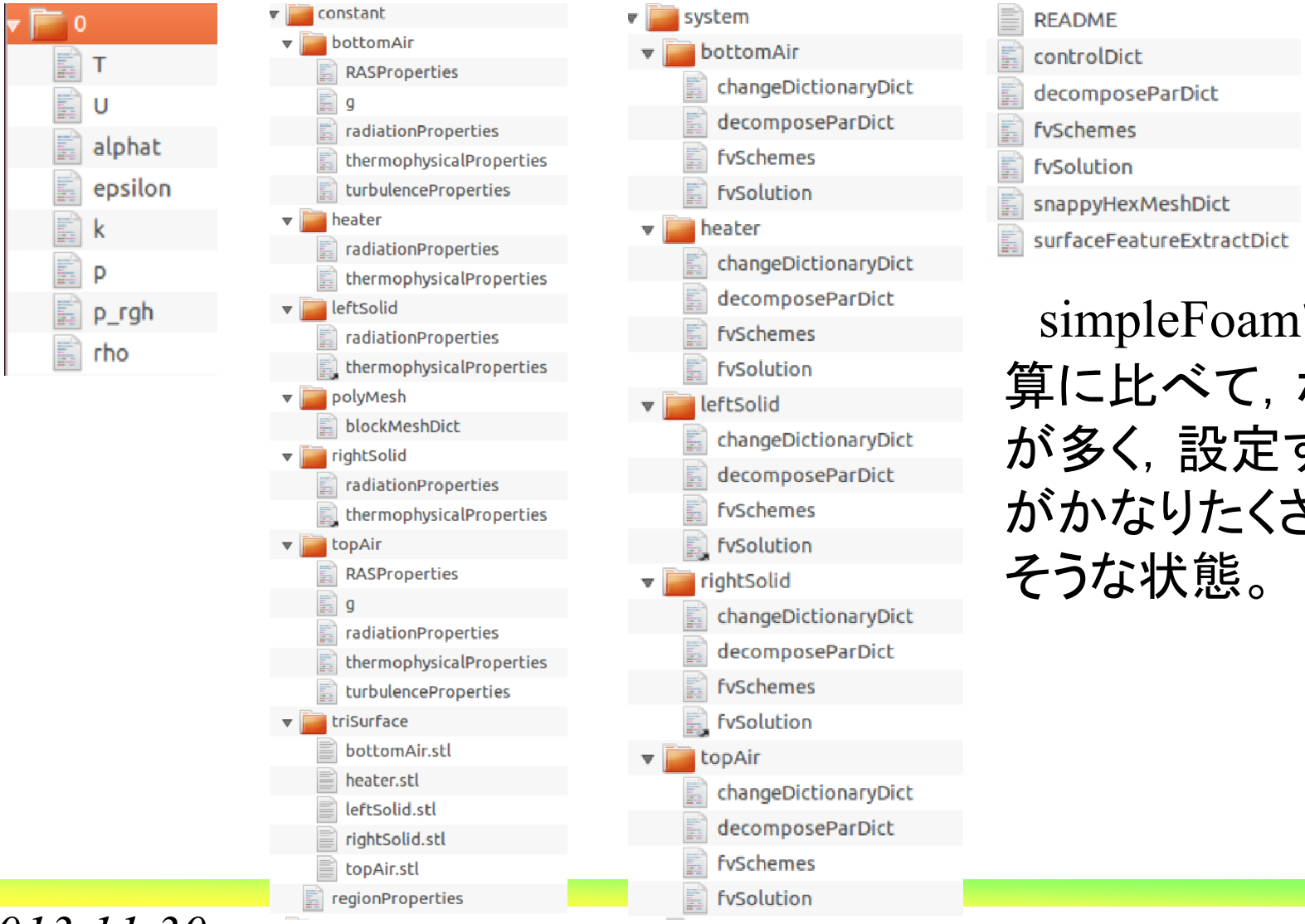


Bounds
X range: -0.1 to -0.0133 (delta: 0.0867)
Y range: -5.78e-19 to 0.008 (delta: 0.008)
Z range: -0.05 to 0.05 (delta: 0.1)



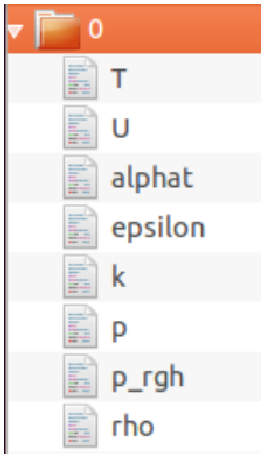
Bounds
X range: 0.0133 to 0.1 (delta: 0.0867)
Y range: -5.78e-19 to 0.008 (delta: 0.008)
Z range: -0.05 to 0.05 (delta: 0.1)

●チュートリアルでの初期状態でのホルダ構成状態



simpleFoamでの計算に比べて，ホルダが多く，設定する事がかなりたくさんありそうな状態。

●初期状態での設定をチェック（0）



T

```
/*--------------------------------*-
| =========                      |
| \\      /  F ield              | OpenFOA
|  \\    /   O peration          | Version
|   \\  /    A nd                | Web:
|    \\/     M anipulation       |
\*--------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      T;
}
// * * * * * * * * * * * * * * * * *

dimensions      [0 0 0 1 0 0 0];

internalField   uniform 300;

boundaryField
{
    ".*"
    {
        type            calculated;
        value           uniform 300;
    }
}
```

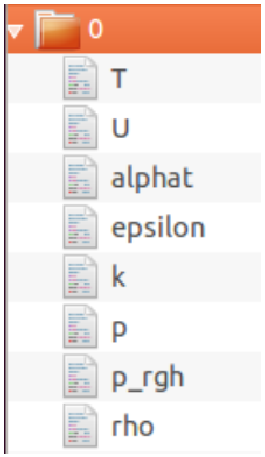
300K

U

```
/*--------------------------------*- C++ -*-
| =========                      |
| \\      /  F ield              | OpenFOAM: The
|  \\    /   O peration          | Version:  2.2.
|   \\  /    A nd                | Web:      www.
|    \\/     M anipulation       |
\*--------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * *

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0.01 0 0);

boundaryField
{
    ".*"
    {
        type            calculated;
        value           uniform (0.01 0 0);
    }
}
```

ダミー設定のような内容になっている

●初期状態での設定をチェック（0）



alphat

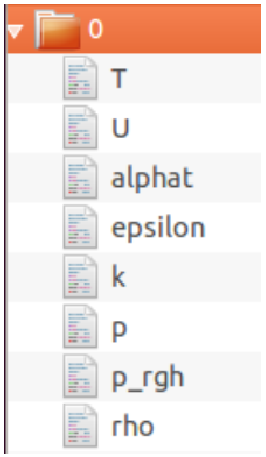
```
/*--------------------------------*-
| =========                      |
| \\      /  F ield              | OpenFOA
|  \\    /   O peration          | Version
|   \\  /    A nd                | Web:
|    \\/     M anipulation       |
\*--------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      alphat;
}
// * * * * * * * * * * * * * * * * * *

dimensions      [1 -1 -1 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    ".*"
    {
        type            calculated;
        value           uniform 0;
    }
}
```

epsilon

```
/*--------------------------------*- C+
| =========                      |
| \\      /  F ield              | OpenFOAM:
|  \\    /   O peration          | Version:
|   \\  /    A nd                | Web:
|    \\/     M anipulation       |
\*--------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      epsilon;
}
// * * * * * * * * * * * * * * * * * *

dimensions      [0 2 -3 0 0 0 0];

internalField   uniform 0.01;

boundaryField
{
    ".*"
    {
        type            calculated;
        value           uniform 0.01;
    }
}
```

ダミー設定のような内容になっている

●初期状態での設定をチェック(0)



k

```
/*--------------------------------*- C
| =========                      |
| \\      /  F ield              | OpenFOAM
|  \\    /   O peration          | Version:
|   \\  /    A nd                | Web:
|    \\/     M anipulation       |
\*---------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      k;
}
// * * * * * * * * * * * * * * * * * *

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0.1;

boundaryField
{
    ".*"
    {
        type            calculated;
        value           uniform 0.1;
    }
}
```

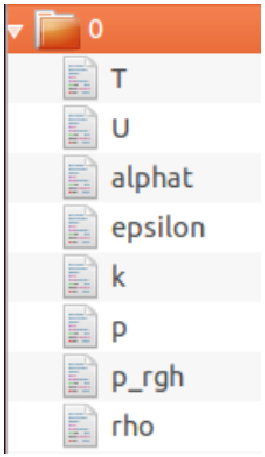
p

```
/*--------------------------------*- C
| =========                      |
| \\      /  F ield              | OpenFOAM
|  \\    /   O peration          | Version:
|   \\  /    A nd                | Web:
|    \\/     M anipulation       |
\*---------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
// * * * * * * * * * * * * * * * * * *

dimensions      [1 -1 -2 0 0 0 0];

internalField   uniform 1e5;

boundaryField
{
    ".*"
    {
        type            calculated;
        value           uniform 1e5;
    }
```

大気圧

ダミー設定のような内容になっている

●初期状態での設定をチェック(0)



p_rgh

```
/*--------------------------------*-
| =========                      |
| \\      /  F ield              | OpenFOA
|  \\    /   O peration          | Version
|   \\  /    A nd                | Web:
|    \\/     M anipulation       |
\*--------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p_rgh;
}
// * * * * * * * * * * * * * * * * * *

dimensions      [1 -1 -2 0 0 0 0];

internalField   uniform 1e5;

boundaryField
{
    ".*"
    {
        type            calculated;
        value           uniform 1e5;
    }
}
```

大気圧

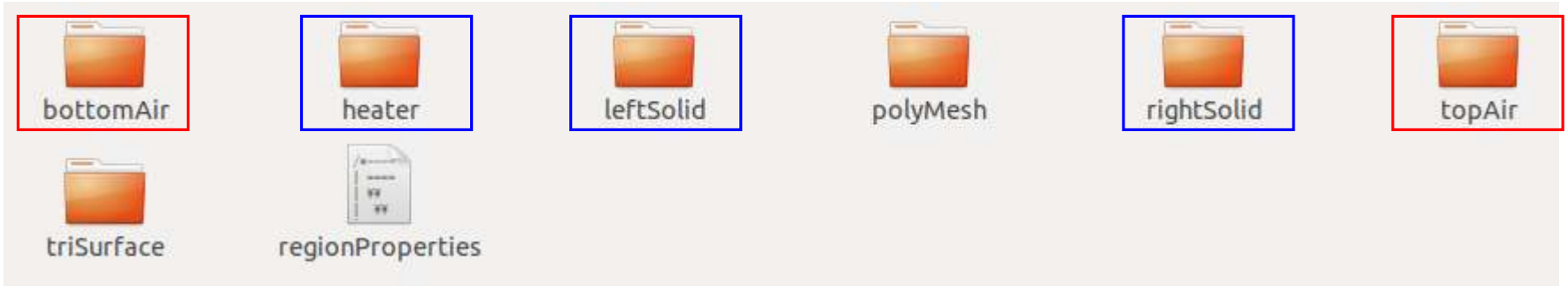
rho

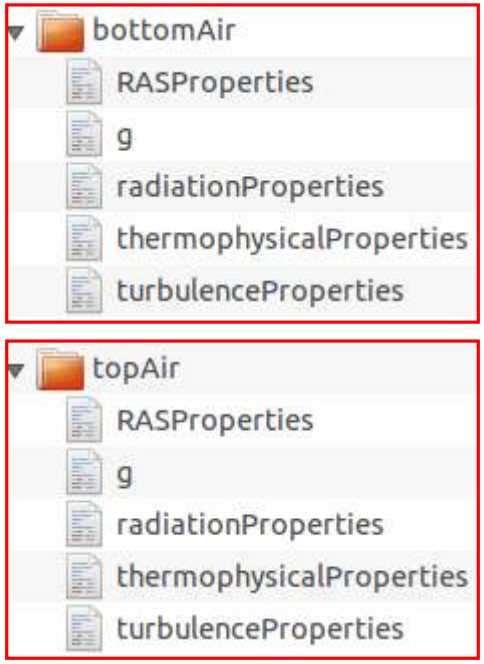
```
/*--------------------------------*- C
| =========                      |
| \\      /  F ield              | OpenFOAM
|  \\    /   O peration          | Version:
|   \\  /    A nd                | Web:
|    \\/     M anipulation       |
\*--------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      rho;
}
// * * * * * * * * * * * * * * * * * *

dimensions      [1 -3 0 0 0 0 0];

internalField   uniform 8000;

boundaryField
{
    ".*"
    {
        type            calculated;
        value           uniform 8000;
    }
}
```

鉄に近い密度

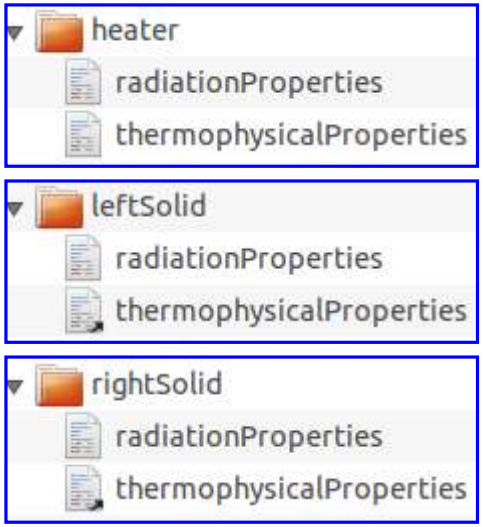
ダミー設定のような内容になっている

●初期状態での設定をチェック（constant）





流体部分



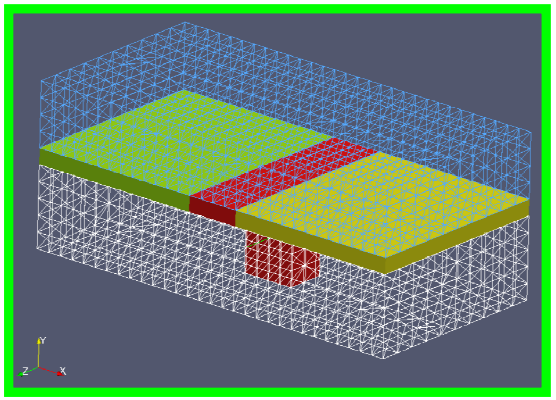
固体部分

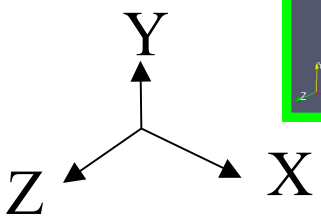
基礎メッシュデータ

形状データ

●初期状態での設定をチェック（constant/polyMesh/blockMesh）

```
convertToMeters 1;

vertices
(
    (-0.1 -0.04  -0.05)
    ( 0.1 -0.04  -0.05)
    ( 0.1  0.04  -0.05)
    (-0.1  0.04  -0.05)
    (-0.1 -0.04   0.05)
    ( 0.1 -0.04   0.05)
    ( 0.1  0.04   0.05)
    (-0.1  0.04   0.05)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (30 10 10) simpleGrading (1 1 1)
);

edges
(
);
```

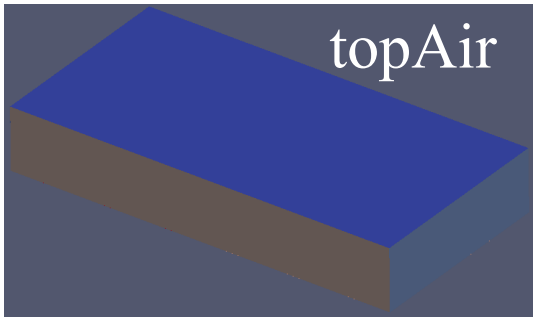


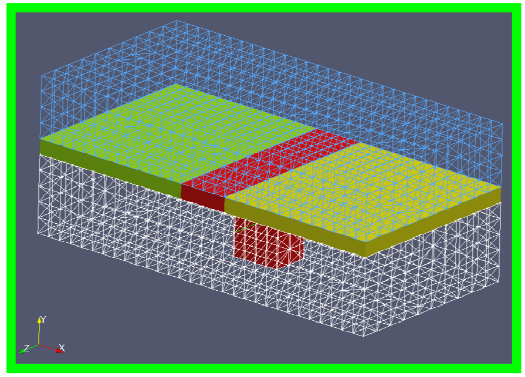


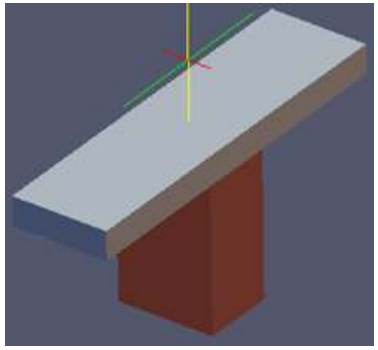

```
boundary
(
    maxY
    {
        type wall;
        faces
        (
            (3 7 6 2)
        );
    }
    minX
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }
    maxX
    {
        type patch;
        faces
        (
            (2 6 5 1)
        );
    }
    minY
    {
        type wall;
        faces
        (
            (1 5 4 0)
        );
    }
    minZ
    {
        type wall;
        faces
        (
            (0 3 2 1)
        );
    }
    maxZ
    {
        type wall;
        faces
        (
            (4 5 6 7)
        );
    }
);

mergePatchPairs
(
);
```

●初期状態での設定をチェック（constant/triSurface）

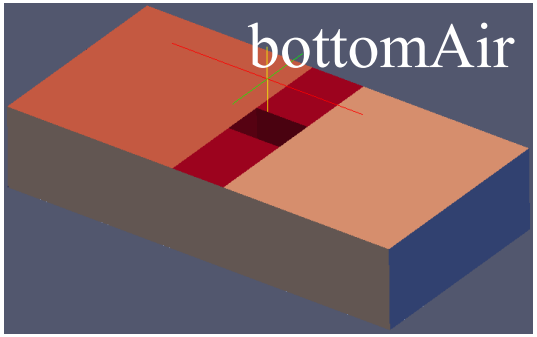
５つのパートがある

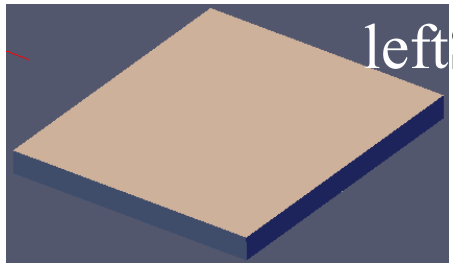


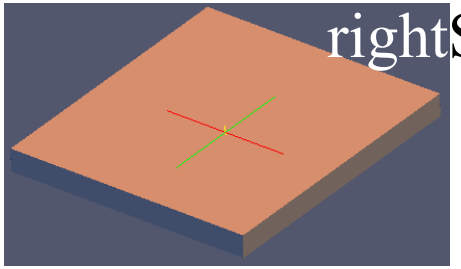




heater







●初期状態での設定をチェック（constant）

regionProperties

```
/*--------------------------------*- C++ -*----
| =========                      |
| \\      /  F ield              | OpenFOAM: The Oper
|  \\    /   O peration          | Version:  2.2.1
|   \\  /    A nd                | Web:      www.Oper
|    \\/     M anipulation       |
\*-----------------------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      regionProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * *

regions
(
    fluid       (bottomAir topAir)
    solid       (heater leftSolid rightSolid)
);
```

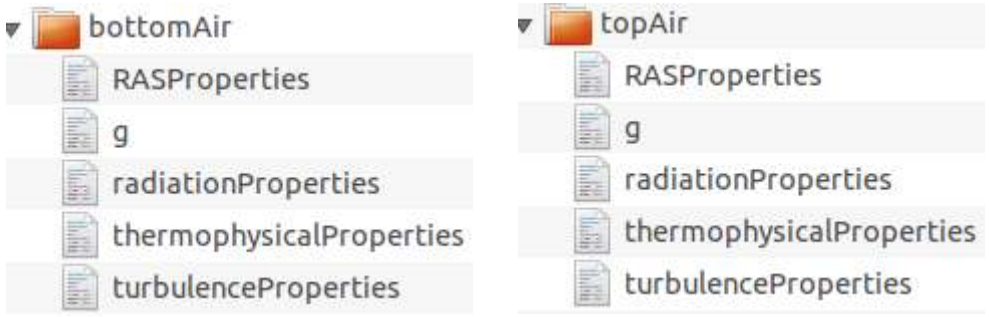
流体部分と固体部分の領域設定

流体部分

固体部分

●初期状態での設定をチェック（constant）

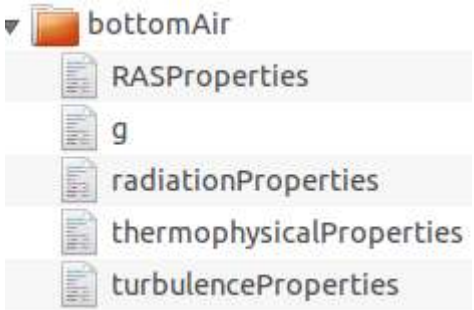
流体部分の物性等の設定



固体部分の物性等の設定

●初期状態での設定をチェック（constant）

流体部分の特性設定（bottomAir，topAir）



同じ特性値が設定

RASProperties　乱流特性

```
/*---------------------------------*
| =========                       |
| \\      /  F ield               | O
|  \\    /   O peration           | V
|   \\  /    A nd                 | W
|    \\/     M anipulation        |
\*---------------------------------*
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      RASProperties;
}
// * * * * * * * * * * * * * * * *

RASModel laminar;

turbulence      on;

printCoeffs     on;
```

層流設定

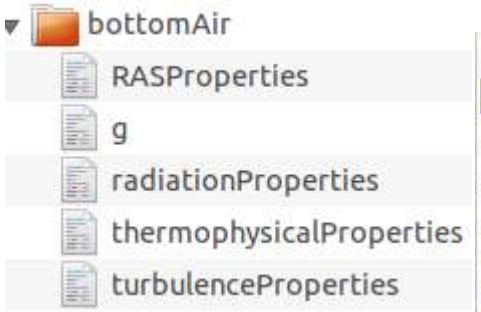
g　重力加速度

```
/*---------------------------------*
| =========                       |
| \\      /  F ield               | OpenF
|  \\    /   O peration           | Versi
|   \\  /    A nd                 | Web:
|    \\/     M anipulation        |
\*---------------------------------*
FoamFile
{
    version     2.0;
    format      ascii;
    class       uniformDimensionedV
    object      g;
}
// * * * * * * * * * * * * * * * * *

dimensions      [0 1 -2 0 0 0 0];
value           (0 -9.81 0);
```

-Y方向

●初期状態での設定をチェック（constant）

流体部分の特性設定（bottomAir，topAir）



同じ特性値が設定

ふく射特性

radiationProperties

```
/*---------------------------------*- (
| =========                       |
| \\      /  F ield               | OpenFOAM
|  \\    /   O peration           | Version:
|   \\  /    A nd                 | Web:
|    \\/     M anipulation        |
\*---------------------------------*
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      radiationProperties;
}
// * * * * * * * * * * * * * * * * * *

radiation       off;

radiationModel  none;
```

ふく射なし

熱物理特性の説明はユーザーガイド7章にある

熱特性

thermophysicalProperties

```
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant/bottomAir";
    object      thermophysicalProperties;
}
// * * * * * * * * * * * * * * * * * *

thermoType
{
    type            heRhoThermo;
    mixture         pureMixture;
    transport       const;
    thermo          hConst;
    equationOfState perfectGas;
    specie          specie;
    energy          sensibleEnthalpy;
}

mixture
{
    specie
    {
        nMoles          1;
        molWeight       28.9;
    }
    thermodynamics
    {
        Cp              1000;
        Hf              0;
    }
    transport
    {
        mu              1.8e-05;
        Pr              0.7;
    }
}
```

空気のモルウェイト

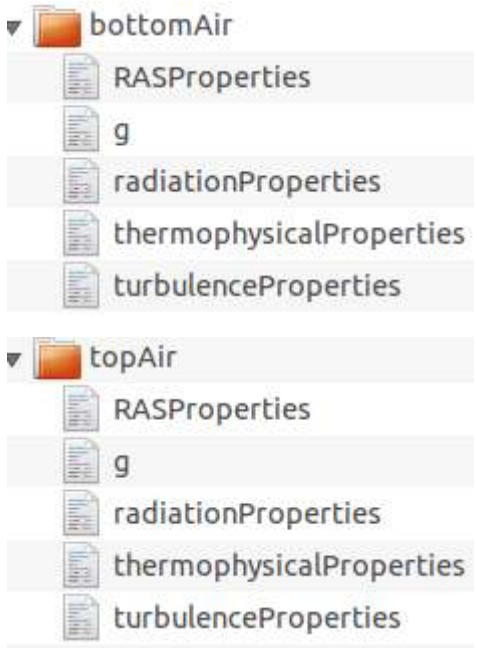
定圧比熱

粘度

プラントル数

●初期状態での設定をチェック（constant）

流体部分の特性設定（bottomAir，topAir）



同じ特性値が設定

turbulenceProperties

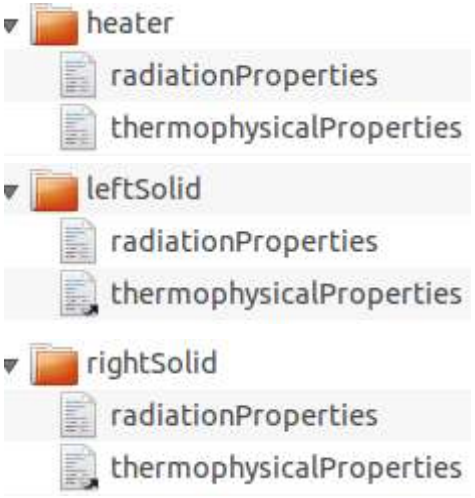
乱流特性

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.2                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      turbulenceProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

simulationType  laminar;
```

層流設定

●初期状態での設定をチェック（constant）

固体部分の特性設定（heater, leftSolid, rightSolid）



同じ特性値が設定

radiationProperties

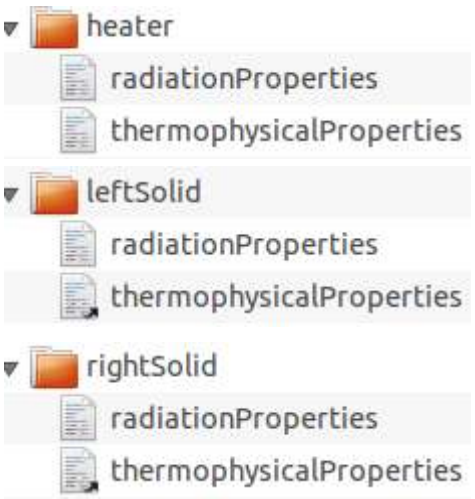
ふく射特性

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/

FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      radiationProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

radiationModel  opaqueSolid;

absorptionEmissionModel constantAbsorptionEmission;

constantAbsorptionEmissionCoeffs
{
    absorptivity    absorptivity [ 0 -1 0 0 0 0 0 ] 0.0;  //opaque
    emissivity      emissivity [ 0 -1 0 0 0 0 0 ] 0.1;
    E               E [ 1 -1 -3 0 0 0 0 ] 0;
}

scatterModel    none;
```

中身の意味は理解できていない

●初期状態での設定をチェック（constant）

固体部分の特性設定（heater, leftSolid, rightSolid）



同じ特性値が設定

中身の意味は理解できていない

thermophysicalProperties

熱特性

```
/*--------------------------------*- C++ -
| =========                      |
| \\      /  F ield              | OpenFOAM: Th
|  \\    /   O peration          | Version:  2.
|   \\  /    A nd                | Web:      ww
|    \\/     M anipulation       |
\*---------------------------------------
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      thermophysicalProperties;
}
// * * * * * * * * * * * * * * * * * * * *

thermoType
{
    type            heSolidThermo;
    mixture         pureMixture;
    transport       constIso;
    thermo          hConst;
    equationOfState rhoConst;
    specie          specie;
    energy          sensibleEnthalpy;
}

mixture
{
    specie
    {
        nMoles      1;
        molWeight   12;
    }
}

transport
{
    kappa   80;
}

thermodynamics
{
    Hf      0;
    Cp      450;
}

equationOfState
{
    rho     8000;
}
```

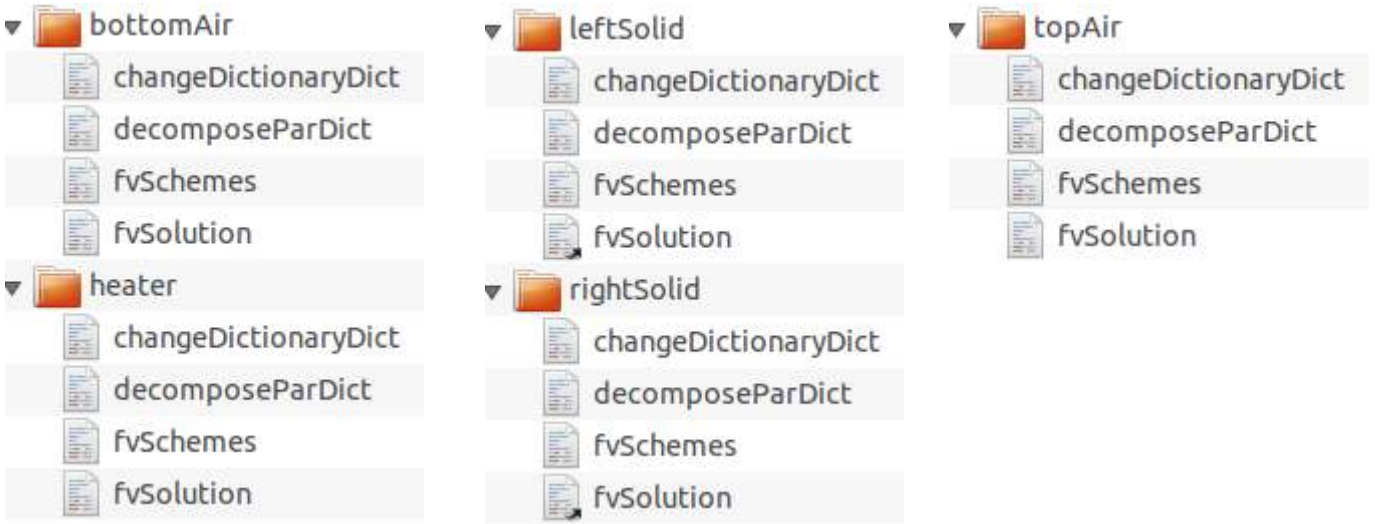
この値は使われない？

鉄の熱伝導率

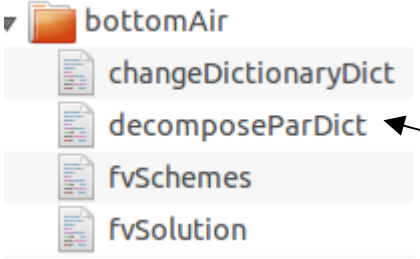
鉄の比熱

鉄に近い密度

●初期状態での設定をチェック（system）



●初期状態での設定をチェック（system/bottomAir）



changeDictionaryコマンドで設定変更をする内容が記述されている

並列計算での分割指定の設定

fvSchemes

```
ddtSchemes
{
    default Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      Gauss upwind;
    div(phi,K)      Gauss linear;
    div(phi,h)      Gauss upwind;
    div(phi,k)      Gauss upwind;
    div(phi,epsilon) Gauss upwind;
    div(phi,R)      Gauss upwind;
    div(R)          Gauss linear;
    div((muEff*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         none;
    laplacian(muEff,U) Gauss linear limited corrected 0.333;
    laplacian(Dp,p_rgh) Gauss linear limited corrected 0.333;
    laplacian(alphaEff,h) Gauss linear limited corrected 0.333;
    laplacian(DkEff,k) Gauss linear limited corrected 0.333;
    laplacian(DepsilonEff,epsilon) Gauss linear limited corrected 0.333;
    laplacian(DREff,R) Gauss linear limited corrected 0.333;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         limited corrected 0.333;
}

fluxRequired
{
    default         no;
    p_rgh;
}
```

●初期状態での設定をチェック（system/bottomAir）

fvSolution

```
solvers
{
    "(rho|rhoFinal)"
    {
        solver          PCG
        preconditioner  DIC;
        tolerance       1e-7;
        relTol          0;
    }

    p_rgh
    {
        solver          GAMG;
        tolerance       1e-7;
        relTol          0.01;

        smoother        GaussSeidel;

        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }

    p_rghFinal
    {
        $p_rgh;
        tolerance       1e-7;
        relTol          0;
    }

    p_rghFinal
    {
        $p_rgh;
        tolerance       1e-7;
        relTol          0;
    }

    "(U|h|k|epsilon|R)"
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-7;
        relTol          0.1;
    }

    "(U|h|k|epsilon|R)Final"
    {
        $U;
        tolerance       1e-07;
        relTol          0;
    }
}

PIMPLE
{
    momentumPredictor   on;
    nCorrectors         2;
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    fields
    {
    }
    equations
    {
        "h.*"           1;
        "U.*"           1;
    }
}
```

●初期状態での設定をチェック（system/bottomAir）

changeDictionaryDict

```
dictionaryReplacement
{
    boundary
    {
        minX
        {
            type            wall;
        }
        maxX
        {
            type            wall;
        }
    }

    U
    {
        internalField   uniform (0.01 0 0);

        boundaryField
        {
            ".*"
            {
                type            fixedValue;
                value           uniform (0 0 0);
            }
        }
    }
```

```
    T
    {
        internalField   uniform 300;

        boundaryField
        {
            ".*"
            {
                type            zeroGradient;
            }

            "bottomAir_to_.*"
            {
                type            compressible::turbulentTemperatureCoupledBaffleMixed;
                neighbourFieldName T;
                kappa           fluidThermo;
                kappaName       none;
                value           uniform 300;
            }
        }
    }
```

層流の計算には利用されない

```
    epsilon
    {
        // Set the value on all bc to non-zero. Not used in simulation
        // since zeroGradient; only used in initialisation.
        internalField   uniform 0.01;

        boundaryField
        {
            ".*"
            {
                type            compressible::epsilonWallFunction;
                value           uniform 0.01;
            }
        }
    }
```

●初期状態での設定をチェック（system/bottomAir）

changeDictionaryDict

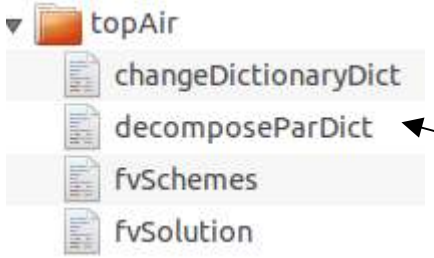
k　層流の計算には利用されない

```
k
{
    internalField   uniform 0.1;

    boundaryField
    {
        ".*"
        {
            type            compressible::kqRWallFunction;
            value           uniform 0.1;
        }
    }
}
```

```
p_rgh
{
    internalField   uniform 1e5;

    boundaryField
    {
        ".*"
        {
            type            fixedFluxPressure;
            value           uniform 1e5;
        }
    }
}
```

```
p
{
    internalField   uniform 1e5;

    boundaryField
    {
        ".*"
        {
            type            calculated;
            value           uniform 1e5;
        }
    }
}
```

●初期状態での設定をチェック（system/topAir）



changeDictionaryコマンドで設定変更をする内容が記述されている

並列計算での分割指定の設定

fvSchemes

```
ddtSchemes
{
    default Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      Gauss upwind;
    div(phi,K)      Gauss linear;
    div(phi,h)      Gauss upwind;
    div(phi,k)      Gauss upwind;
    div(phi,epsilon) Gauss upwind;
    div(phi,R)      Gauss upwind;
    div(R)          Gauss linear;
    div((muEff*dev2(T(grad(U))))) Gauss linear;
}
```

```
laplacianSchemes
{
    default         none;
    laplacian(muEff,U) Gauss linear limited corrected 0.333;
    laplacian(Dp,p_rgh) Gauss linear limited corrected 0.333;
    laplacian(alphaEff,h) Gauss linear limited corrected 0.333;
    laplacian(DkEff,k) Gauss linear limited corrected 0.333;
    laplacian(DepsilonEff,epsilon) Gauss linear limited corrected 0.333;
    laplacian(DREff,R) Gauss linear limited corrected 0.333;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         limited corrected 0.333;
}

fluxRequired
{
    default         no;
    p_rgh;
}
```

●初期状態での設定をチェック（system/topAir）

fvSolution

```
solvers
{
    "(rho|rhoFinal)"
    {
        solver          PCG
        preconditioner  DIC;
        tolerance       1e-7;
        relTol          0;
    }

    p_rgh
    {
        solver          GAMG;
        tolerance       1e-7;
        relTol          0.01;

        smoother        GaussSeidel;

        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;

        maxIter         100;
    }

    p_rghFinal
    {
        $p_rgh;
        tolerance       1e-7;
        relTol          0;
    }

    p_rghFinal
    {
        $p_rgh;
        tolerance       1e-7;
        relTol          0;
    }

    "(U|h|k|epsilon|R)"
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-7;
        relTol          0.1;
    }

    "(U|h|k|epsilon|R)Final"
    {
        $U;
        tolerance       1e-07;
        relTol          0;
    }
}

PIMPLE
{
    momentumPredictor   on;
    nCorrectors         2;
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    fields
    {
    }
    equations
    {
        "h.*"           1;
        "U.*"           1;
    }
}
```

bottomAirにはない

●初期状態での設定をチェック（system/topAir）

changeDictionaryDict

```
dictionaryReplacement
{
    U
    {
        internalField   uniform (0.1 0 0);

        boundaryField
        {
            ".*"
            {
                type            fixedValue;
                value           uniform (0 0 0);
            }
            minX
            {
                type            fixedValue;
                value           uniform ( 0.1 0 0 );
            }
            maxX
            {
                type            inletOutlet;
                inletValue      uniform ( 0 0 0 );
                value           uniform ( 0.1 0 0 );
            }
        }
    }
```

●初期状態での設定をチェック（system/topAir）

changeDictionaryDict

T

```
{
    internalField   uniform 300;

    boundaryField
    {
        ".*"
        {
            type            zeroGradient;
        }

        minX
        {
            type            fixedValue;
            value           uniform 300;
        }
        maxX
        {
            type            inletOutlet;
            inletValue      uniform 300;
            value           uniform 300;
        }

        "topAir_to_.*"
        {
            type            compressible::turbulentTemperatureCoupledBaffleMixed;
            neighbourFieldName T;
            kappa           fluidThermo;
            kappaName       none;
            value           uniform 300;
        }
    }
}
```

●初期状態での設定をチェック（system/topAir）

changeDictionaryDict

```
epsilon
{
    internalField   uniform 0.01;

    boundaryField
    {
        ".*"
        {
            type            compressible::epsilonWallFunction;
            value           uniform 0.01;
        }

        minX
        {
            type            fixedValue;
            value           uniform 0.01;
        }
        maxX
        {
            type            inletOutlet;
            inletValue      uniform 0.01;
            value           uniform 0.01;
        }
    }
}
```

```
k
{
    internalField   uniform 0.1;

    boundaryField
    {
        ".*"
        {
            type            compressible::kqRWallFunction;
            value           uniform 0.1;
        }

        minX
        {
            type            fixedValue;
            value           uniform 0.1;
        }
        maxX
        {
            type            inletOutlet;
            inletValue      uniform 0.1;
            value           uniform 0.1;
        }
    }
}
```

層流の計算には利用されない

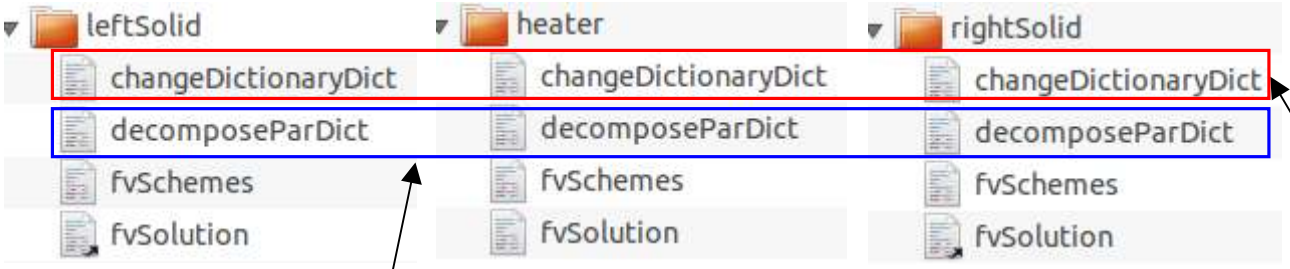
●初期状態での設定をチェック（system/topAir）

changeDictionaryDict

```
p_rgh
{
    internalField   uniform 1e5;

    boundaryField
    {
        ".*"
        {
            type            fixedFluxPressure;
            value           uniform 1e5;
        }

        maxX
        {
            type            fixedValue;
            value           uniform 1e5;
        }
    }
}
```

```
p
{
    internalField   uniform 1e5;

    boundaryField
    {
        ".*"
        {
            type            calculated;
            value           uniform 1e5;
        }

        maxX
        {
            type            calculated;
            value           uniform 1e5;
        }
    }
}
```

●初期状態での設定をチェック（system/leftSolid,rightSolid,heater）



並列計算での分割指定の設定

changeDictionaryコマンドで設定変更をする内容が記述されている。ただし，個々の設定は異なる。

fvSchemes

```
ddtSchemes
{
    default Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;
}

laplacianSchemes
{
    default             none;
    laplacian(alpha,h)  Gauss linear limited corrected 0.333;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         limited corrected 0.333;
}

fluxRequired
{
    default         no;
}
```

●初期状態での設定をチェック（system/leftSolid,rightSolid,heater）

fvSolution

```
solvers
{
    h
    {
        solver           PCG;
        preconditioner   DIC;
        tolerance        1e-06;
        relTol           0.1;
    }

    hFinal
    {
        $h;
        tolerance        1e-06;
        relTol           0;
    }
}

PIMPLE
{
    nNonOrthogonalCorrectors 1;
}
```

●初期状態での設定をチェック（system/leftSolid）

changeDictionaryDict

```
dictionaryReplacement
{
    boundary                        T
    {                               {
        minZ                            internalField   uniform 300;
        {
            type            patch;      boundaryField
        }                               {
        maxZ                                ".*"
        {                                   {
            type            patch;              type            zeroGradient;
        }                                       value           uniform 300;
    }                                       }
                                            "leftSolid_to_.*"
                                            {
                                                type            compressible::turbulentTemperatureCoupledBaffleMixed;
                                                neighbourFieldName T;
                                                kappa           solidThermo;
                                                kappaName       none;
                                                value           uniform 300;
                                            }
                                        }
                                    }
                                }
```

●初期状態での設定をチェック（system/rightSolid）

changeDictionaryDict

```
dictionaryReplacement
{
    boundary                        T
    {                               {
        minZ                            internalField   uniform 300;
        {
            type            patch;      boundaryField
        }                               {
        maxZ                                ".*"
        {                                   {
            type            patch;              type            zeroGradient;
        }                                       value           uniform 300;
    }                                       }
                                            "rightSolid_to_.*"
                                            {
                                                type            compressible::turbulentTemperatureCoupledBaffleMixed;
                                                neighbourFieldName T;
                                                kappa           solidThermo;
                                                kappaName       none;
                                                value           uniform 300;
                                            }
                                        }
                                    }
                                }
```

●初期状態での設定をチェック（system/heater）

changeDictionaryDict

```
dictionaryReplacement
{
    boundary
    {
        minY
        {
            type            patch;
        }
        minZ
        {
            type            patch;
        }
        maxZ
        {
            type            patch;
        }
    }
```

```
    T
    {
        internalField   uniform 300;

        boundaryField
        {
            ".*"
            {
                type            zeroGradient;
                value           uniform 300;
            }
            "heater_to_.*"
            {
                type            compressible::turbulentTemperatureCoupledBaffleMixed;
                neighbourFieldName T;
                kappa           solidThermo;
                kappaName       none;
                value           uniform 300;
            }
            minY
            {
                type            fixedValue;
                value           uniform 500;
            }
        }
    }
```

熱源温度設定

●計算手順を知るためにAllrunを順番に見ていく

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


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

rm -rf constant/polyMesh/sets

runApplication blockMesh
runApplication surfaceFeatureExtract

runApplication snappyHexMesh -overwrite
runApplication splitMeshRegions -cellZones -overwrite
```

基礎メッシュの作成

特徴線の取出し

sHMによるメッシュ作成

メッシュを複数領域に分割
マルチリージョンを扱うに
は必要な設定

```
# remove fluid fields from solid regions (important for post-processing)
for i in heater leftSolid rightSolid
do
   rm -f 0*/$i/{mut,alphat,epsilon,k,U,p_rgh}
done
```

3つのソリッド領域

ソリッド領域に関係ないものを消す

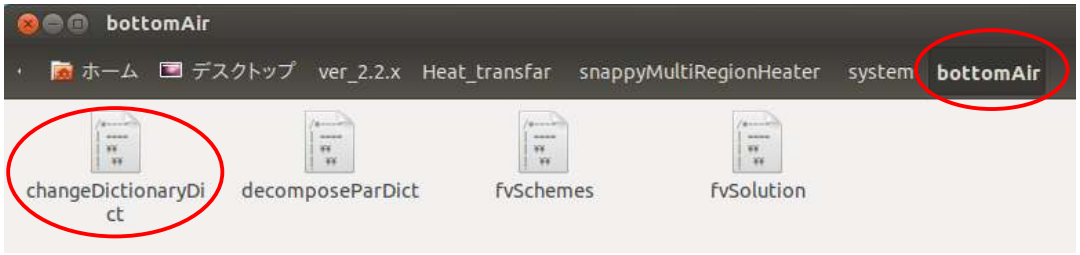
ただし，この部分は上手く動きません。
rmのコマンドで，引数が-fになっているためエラーが端末にバックされませんが
-fをなしにしてAllrunを実施しますとエラーが出ます。
ポストプロセスに重要とコメントが入っているが。。。

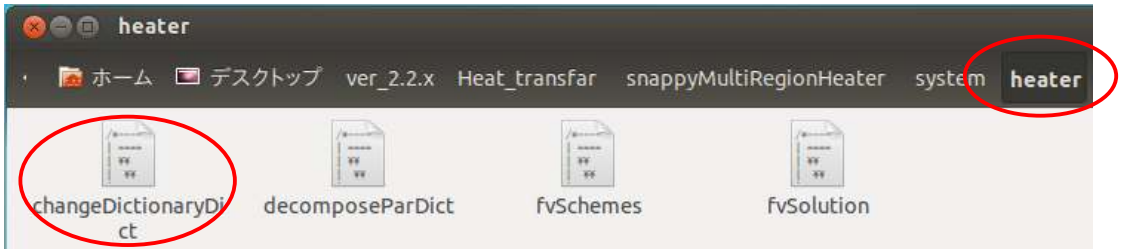
rm: `0*/heater/{mut,alphat,epsilon,k,U,p_rgh}‘ を削除できません:
そのようなファイルやディレクトリはありません
rm: `0*/leftSolid/{mut,alphat,epsilon,k,U,p_rgh}‘ を削除できません:
そのようなファイルやディレクトリはありません
rm: `0*/rightSolid/{mut,alphat,epsilon,k,U,p_rgh}‘ を削除できません:
そのようなファイルやディレクトリはありません

```
for i in bottomAir topAir heater leftSolid rightSolid
do
    changeDictionary -region $i > log.changeDictionary.$i 2>&1
done
```

全部の領域

changeDictonaryDictに基づいて修正を行う






```
dictionaryReplacement
{
    boundary
    {
        minX
        {
            type            wall;
        }
        maxX
        {
            type            wall;
        }
    }

    U
    {
        internalField   uniform (0.01 0 0);

        boundaryField
        {
            ".*"
            {
                type            fixedValue;
                value           uniform (0 0 0);
            }
        }
    }
```

境界，初期値の修正

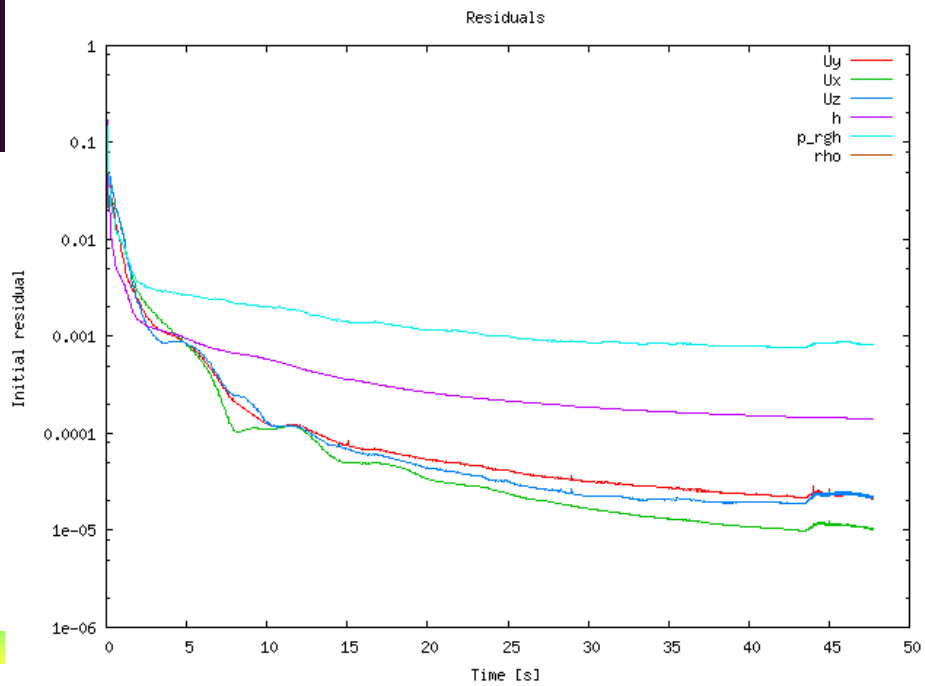
```
#-- Run on single processor
runApplication `getApplication`

echo
echo "creating files for paraview post-processing"
echo
paraFoam -touchAll
```

計算の実施

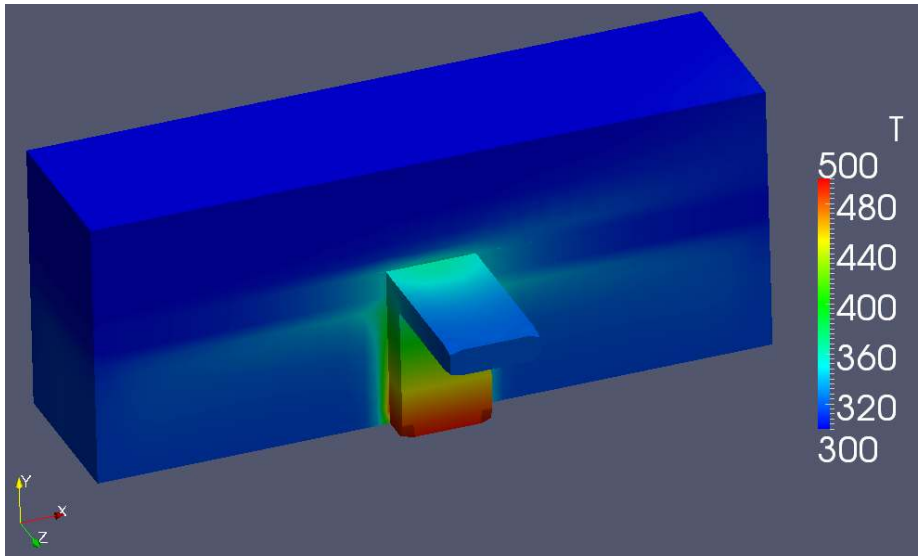
計算結果の用意

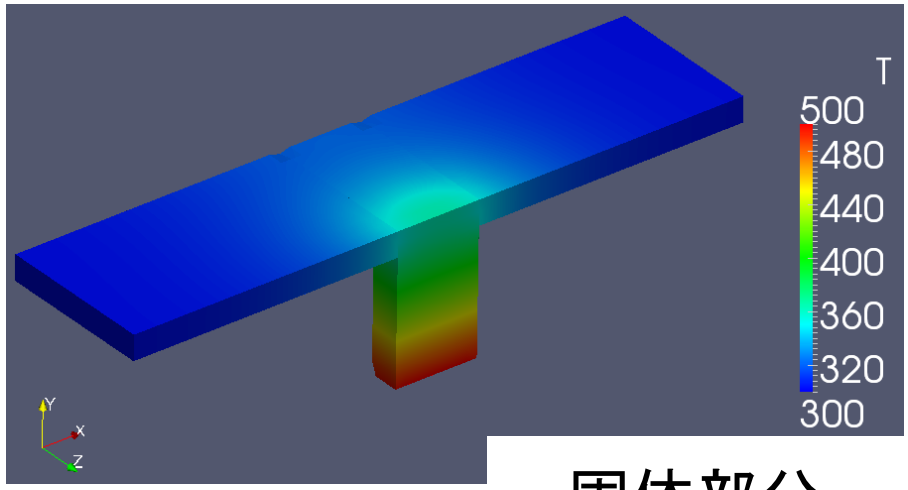
実際に動かすとこんな感じになる

```
sakuramaru@SAKURA-MARU:~/Desktop/ver_2.2.x/Heat_transfar/snappyMultiRegionHeater$ ./Allrun
Running blockMesh on /home/sakuramaru/Desktop/ver_2.2.x/Heat_transfar/snappyMultiRegionHeater
Running surfaceFeatureExtract on /home/sakuramaru/Desktop/ver_2.2.x/Heat_transfar/snappyMultiRegionHeater
Running snappyHexMesh on /home/sakuramaru/Desktop/ver_2.2.x/Heat_transfar/snappyMultiRegionHeater
Running splitMeshRegions on /home/sakuramaru/Desktop/ver_2.2.x/Heat_transfar/snappyMultiRegionHeater
Running chtMultiRegionFoam on /home/sakuramaru/Desktop/ver_2.2.x/Heat_transfar/snappyMultiRegionHeater

creating files for paraview post-processing

created 'snappyMultiRegionHeater.blockMesh'
created 'snappyMultiRegionHeater.OpenFOAM'
created 'snappyMultiRegionHeater{bottomAir}.OpenFOAM'
created 'snappyMultiRegionHeater{heater}.OpenFOAM'
created 'snappyMultiRegionHeater{leftSolid}.OpenFOAM'
created 'snappyMultiRegionHeater{rightSolid}.OpenFOAM'
created 'snappyMultiRegionHeater{topAir}.OpenFOAM'
```

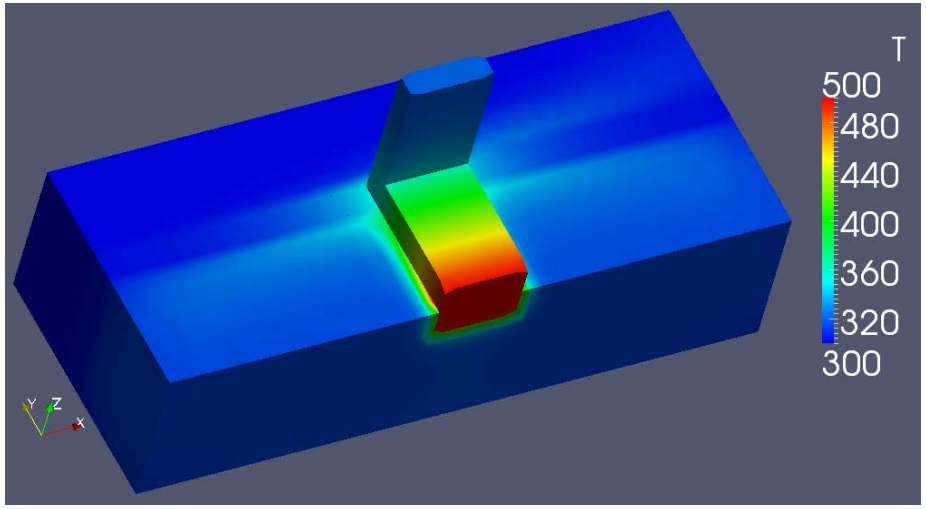


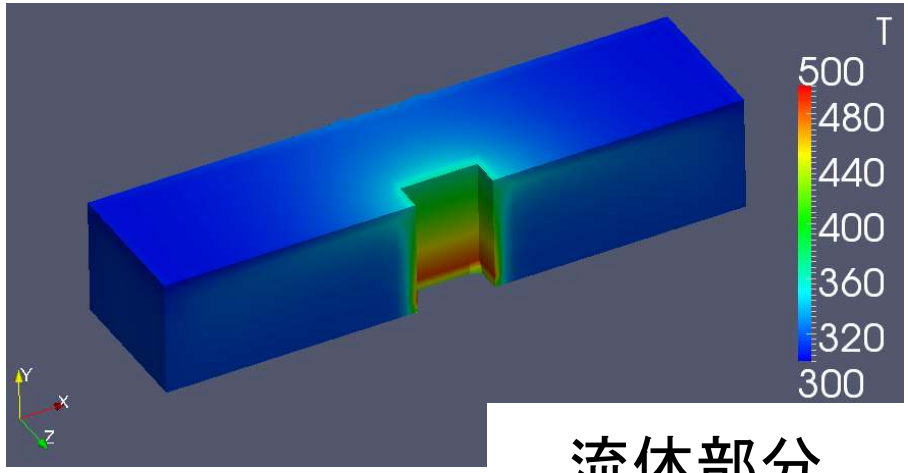
●計算結果はどうなっているか？　温度分布





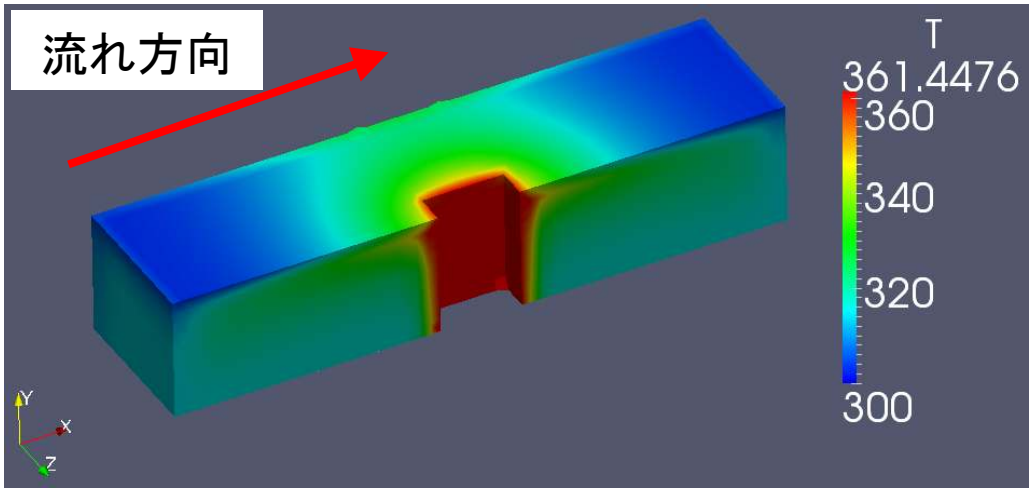
固体部分



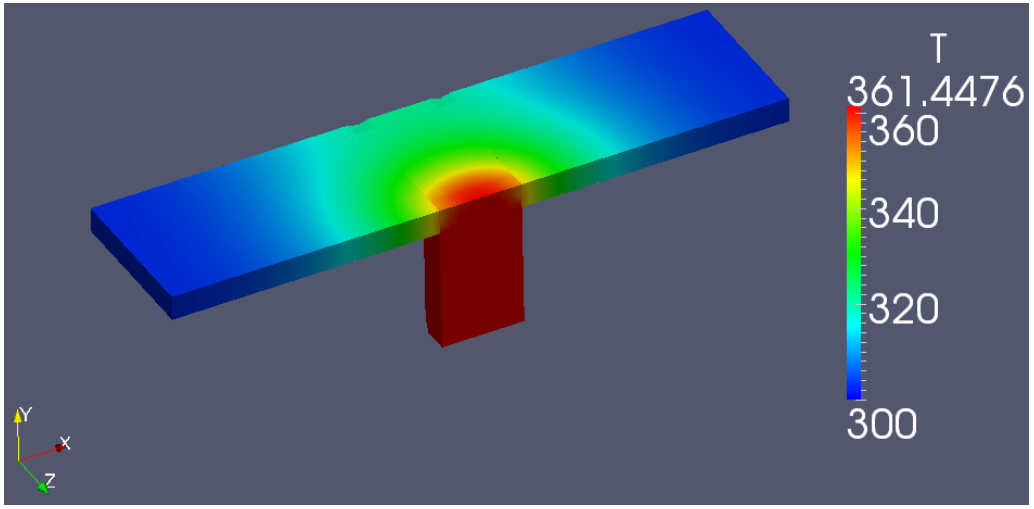


流体部分

●計算結果はどうなっているか？　温度分布

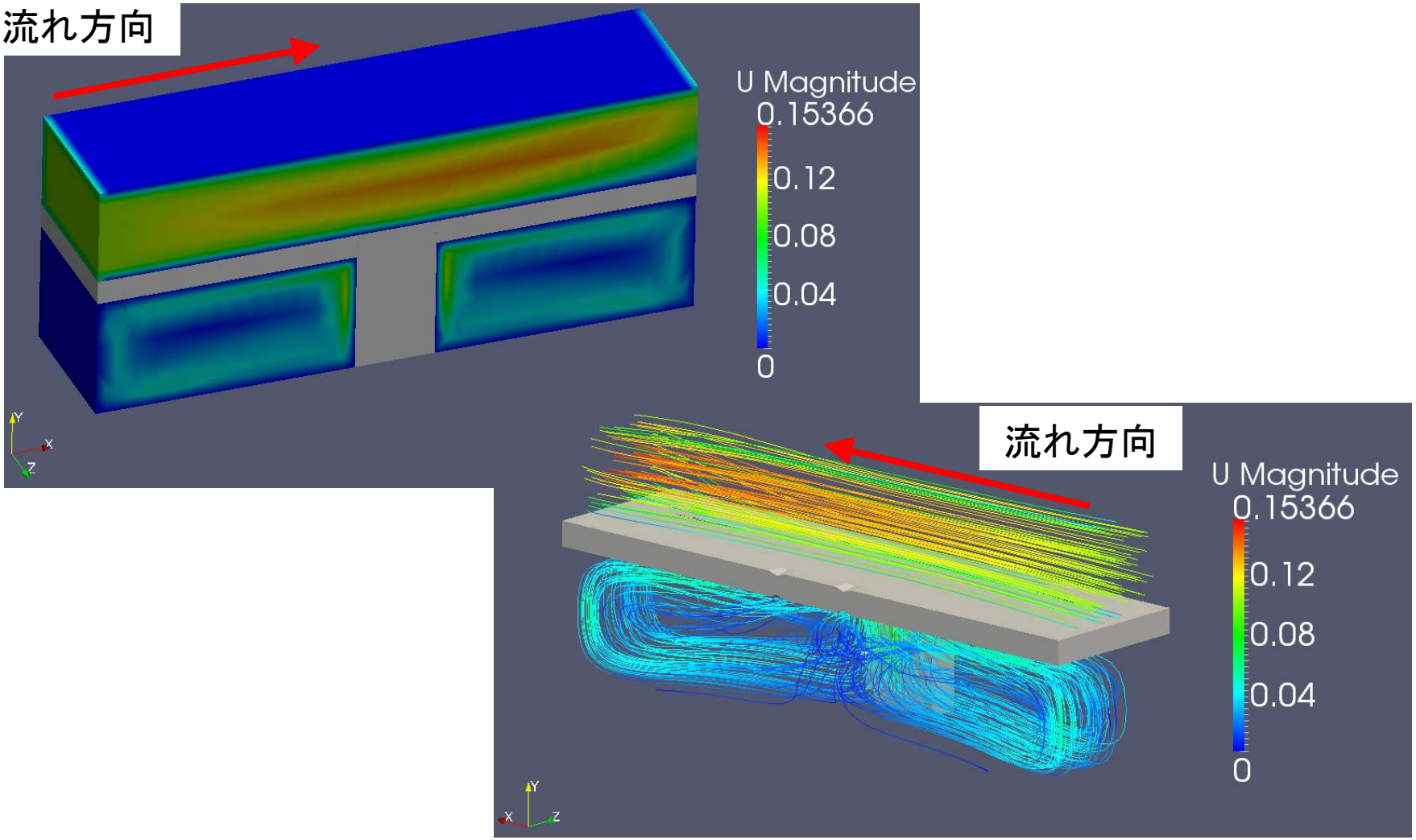


流体部分



固体部分

●計算結果はどうなっているか？　速度分布



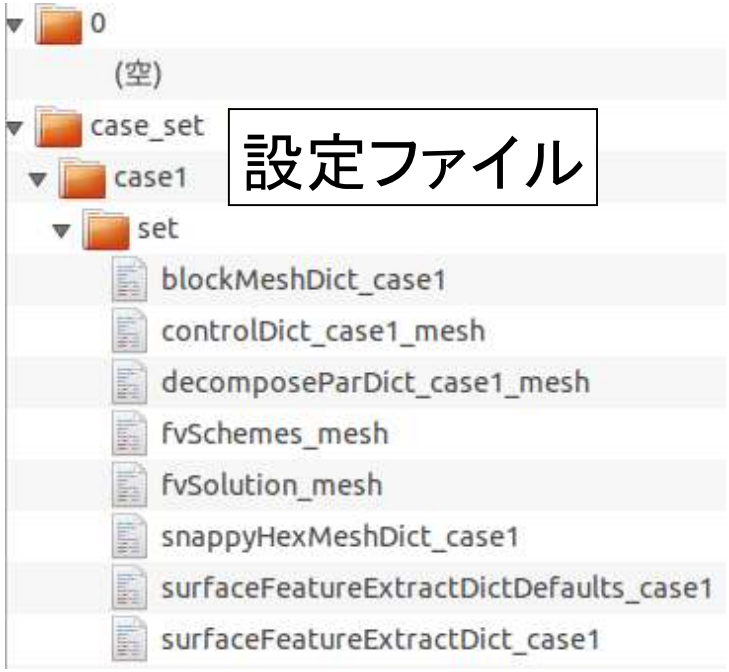
2. コマンドによるモデル作成と設定

●どうやってモデルを作成するか？

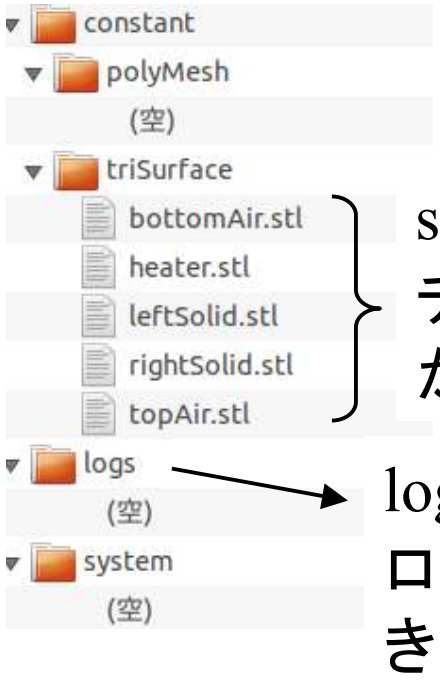
チュートリアルはAllrun一発で走るが，モデルをどうやって構築していくかが問題である。チュートリアルでは，計算が動くように色々と設定がされているが，全くの素の状態からのモデル作りをやってみる。

今回はマクロを作成する事で，モデルの設定，計算をやってみました。出来れば，TreeFoamを使ってスマートにやっていきたいのですが，操作手順が良くわかりません。ぜひやってみた方は手順を教えて下さい。

●メッシュ作成に必要なファイルの設定



設定ファイル



stlファイル
チュートリアル
から流用

logsフォルダは
ログファイル書
き出し用

case_setのホルダはメッシュ作成に必要なファイルを入れておく

初期ホルダは事前に設定しておく

●コマンドによるメッシュ作成手順

①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
SNAPPYHEXMESH_FEATURE=surfaceFeatureExtractDict_case1
SNAPPYHEXMESH_DEFAULT=surfaceFeatureExtractDictDefaults_case1
CONTROL_DICT=controlDict_case1_mesh
FVSCHEMES_MESH=fvSchemes_mesh
FVSOLUTION_MESH=fvSolution_mesh

REGION_FLOW_1=bottomAir
REGION_FLOW_2=topAir
REGION_SOLID_1=heater
REGION_SOLID_2=leftSolid
REGION_SOLID_3=rightSolid
```

流体，固体の各領域名の設定
constant，systemのホルダ設定に利用

②基礎メッシュの作成（blockMesh）

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

blockMeshを実施するために必要なファイルを設定

blockMeshの実施

③特徴線の取出し（surfaceFeatureExtract）

```
cp -r  $CASE_DIR/$SNAPPYHEXMESH_FEATURE system/surfaceFeatureExtractDict
cp -r  $CASE_DIR/$SNAPPYHEXMESH_DEFAULT system/surfaceFeatureExtractDictDefaults
$runApplication surfaceFeatureExtract > logs/log.surfaceFeatureExtract
```

surfaceFeatureExtractの実施

surfaceFeatureExtractを実施するために必要なファイルを設定

④メッシュの並列処理のための前準備（decomposePar）

```
cp -r $CASE_DIR/$DECOMPOSE_PAR system/decomposeParDict
$runApplication decomposePar > logs/log.decomposePar
```

decomposeParの実施

decomposeParを実施するために必要なファイルを設定

⑤並列メッシュに必要なファイルのコピー

```
cp -r constant/triSurface/ processor0/constant/
cp -r constant/triSurface/ processor1/constant/
cp -r constant/triSurface/ processor2/constant/
cp -r constant/triSurface/ processor3/constant/

mkdir processor0/system
mkdir processor1/system
mkdir processor2/system
mkdir processor3/system

cp -r $CASE_DIR/$FVSCHEMES_MESH processor0/system/fvSchemes
cp -r $CASE_DIR/$FVSCHEMES_MESH processor1/system/fvSchemes
cp -r $CASE_DIR/$FVSCHEMES_MESH processor2/system/fvSchemes
cp -r $CASE_DIR/$FVSCHEMES_MESH processor3/system/fvSchemes

cp -r $CASE_DIR/$FVSOLUTION_MESH processor0/system/fvSolution
cp -r $CASE_DIR/$FVSOLUTION_MESH processor1/system/fvSolution
cp -r $CASE_DIR/$FVSOLUTION_MESH processor2/system/fvSolution
cp -r $CASE_DIR/$FVSOLUTION_MESH processor3/system/fvSolution
```

4並列でメッシュを作成するため，それに合うように準備

⑥並列でメッシュの作成（snappyHexMesh）

```
cp -r $CASE_DIR/$SNAPPYHEXMESH_DICT system/snappyHexMeshDict
$runApplication mpirun -np 4 snappyHexMesh -parallel > logs/log.snappyHexMesh
```

snappyHexMeshを4並列で実施

snappyHexMeshを実施するために必要なファイルを設定

⑦リージョンを分けの実施

```
$runApplication mpirun -np 4 splitMeshRegions -cellZones -parallel
                                -overwrite > logs/log.splitMeshRegions
```

⑧メッシュの再構築（reconstructParMesh）

```
$runApplication reconstructParMesh -time 2 -mergeTol 1e-6 > logs/log.reconstructParMesh
$runApplication reconstructParMesh -time 2 -mergeTol 1e-6 -region $REGION_FLOW_1 >  logs/log.reconstructParMesh_$REGION_FLOW_1
$runApplication reconstructParMesh -time 2 -mergeTol 1e-6 -region $REGION_FLOW_2 >  logs/log.reconstructParMesh_$REGION_FLOW_2
$runApplication reconstructParMesh -time 2 -mergeTol 1e-6 -region $REGION_SOLID_1 > logs/log.reconstructParMesh_$REGION_SOLID_1
$runApplication reconstructParMesh -time 2 -mergeTol 1e-6 -region $REGION_SOLID_2 > logs/log.reconstructParMesh_$REGION_SOLID_2
$runApplication reconstructParMesh -time 2 -mergeTol 1e-6 -region $REGION_SOLID_3 > logs/log.reconstructParMesh_$REGION_SOLID_3

rm -r processor0
rm -r processor1
rm -r processor2
rm -r processor3
```

必要の無くなったホルダを削除

⑨メッシュ状態のチェック（checkMesh）

```
$runApplication checkMesh -time 2 -allGeometry > logs/log.checkMesh
$runApplication checkMesh -time 2 -allGeometry -region $REGION_FLOW_1 > logs/log.checkMesh_$REGION_FLOW_1
$runApplication checkMesh -time 2 -allGeometry -region $REGION_FLOW_2 > logs/log.checkMesh_$REGION_FLOW_2
$runApplication checkMesh -time 2 -allGeometry -region $REGION_SOLID_1 > logs/log.checkMesh_$REGION_SOLID_1
$runApplication checkMesh -time 2 -allGeometry -region $REGION_SOLID_2 > logs/log.checkMesh_$REGION_SOLID_2
$runApplication checkMesh -time 2 -allGeometry -region $REGION_SOLID_3 > logs/log.checkMesh_$REGION_SOLID_3
```

メッシュの状態は計算収束に影響があるためチェックは必要

⑩メッシュのリナンバリング（renumberMesh）

各ホルダごとにリナンバリング

```
mkdir system/$REGION_FLOW_1
mkdir system/$REGION_FLOW_2
mkdir system/$REGION_SOLID_1
mkdir system/$REGION_SOLID_2
mkdir system/$REGION_SOLID_3
```

リナンバリングに必要な
ホルダとファイルを用意

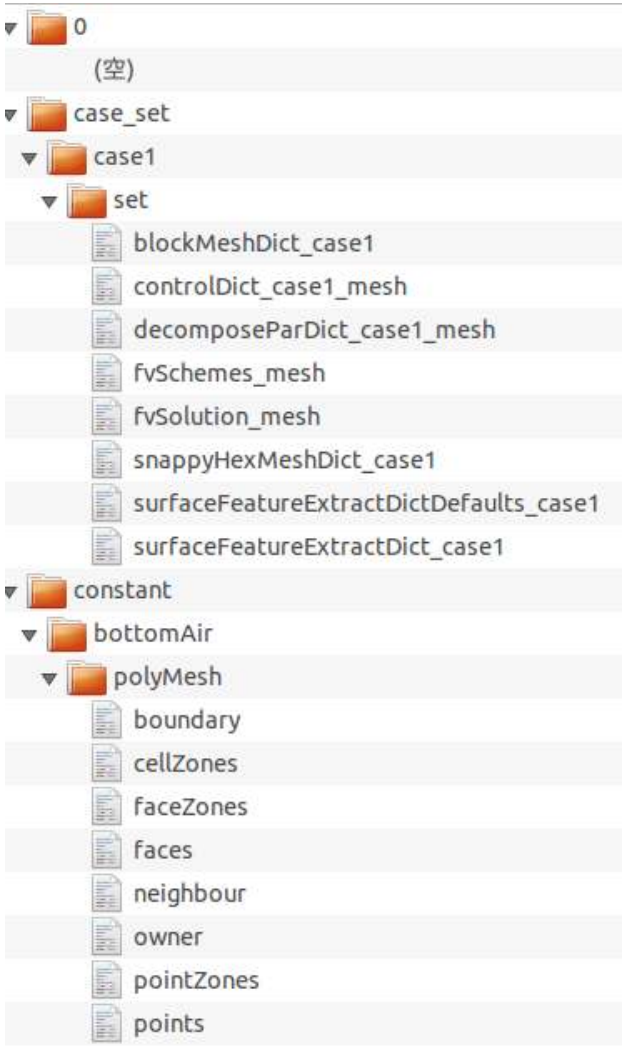
```
cp -r $CASE_DIR/$FVSCHEMES_MESH  system/fvSchemes
cp -r $CASE_DIR/$FVSCHEMES_MESH  system/$REGION_FLOW_1/fvSchemes
cp -r $CASE_DIR/$FVSCHEMES_MESH  system/$REGION_FLOW_2/fvSchemes
cp -r $CASE_DIR/$FVSCHEMES_MESH  system/$REGION_SOLID_1/fvSchemes
cp -r $CASE_DIR/$FVSCHEMES_MESH  system/$REGION_SOLID_2/fvSchemes
cp -r $CASE_DIR/$FVSCHEMES_MESH  system/$REGION_SOLID_3/fvSchemes
```

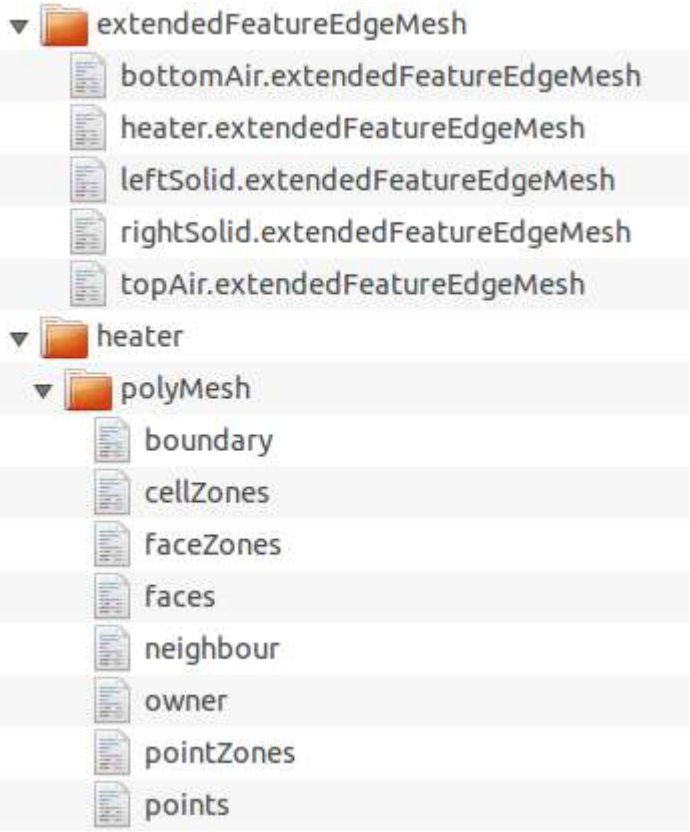
```
cp -r $CASE_DIR/$FVSOLUTION_MESH system/fvSolution
cp -r $CASE_DIR/$FVSOLUTION_MESH system/$REGION_FLOW_1/fvSolution
cp -r $CASE_DIR/$FVSOLUTION_MESH system/$REGION_FLOW_2/fvSolution
cp -r $CASE_DIR/$FVSOLUTION_MESH system/$REGION_SOLID_1/fvSolution
cp -r $CASE_DIR/$FVSOLUTION_MESH system/$REGION_SOLID_2/fvSolution
cp -r $CASE_DIR/$FVSOLUTION_MESH system/$REGION_SOLID_3/fvSolution

$runApplication renumberMesh -time 2 > logs/log.renumberMesh
$runApplication renumberMesh -time 2 -region $REGION_FLOW_1 > logs/log.renumberMesh_$REGION_FLOW_1
$runApplication renumberMesh -time 2 -region $REGION_FLOW_2 > logs/log.renumberMesh_$REGION_FLOW_2
$runApplication renumberMesh -time 2 -region $REGION_SOLID_1 > logs/log.renumberMesh_$REGION_SOLID_1
$runApplication renumberMesh -time 2 -region $REGION_SOLID_2 > logs/log.renumberMesh_$REGION_SOLID_2
$runApplication renumberMesh -time 2 -region $REGION_SOLID_3 > logs/log.renumberMesh_$REGION_SOLID_3
```

⑪ホルダの調整

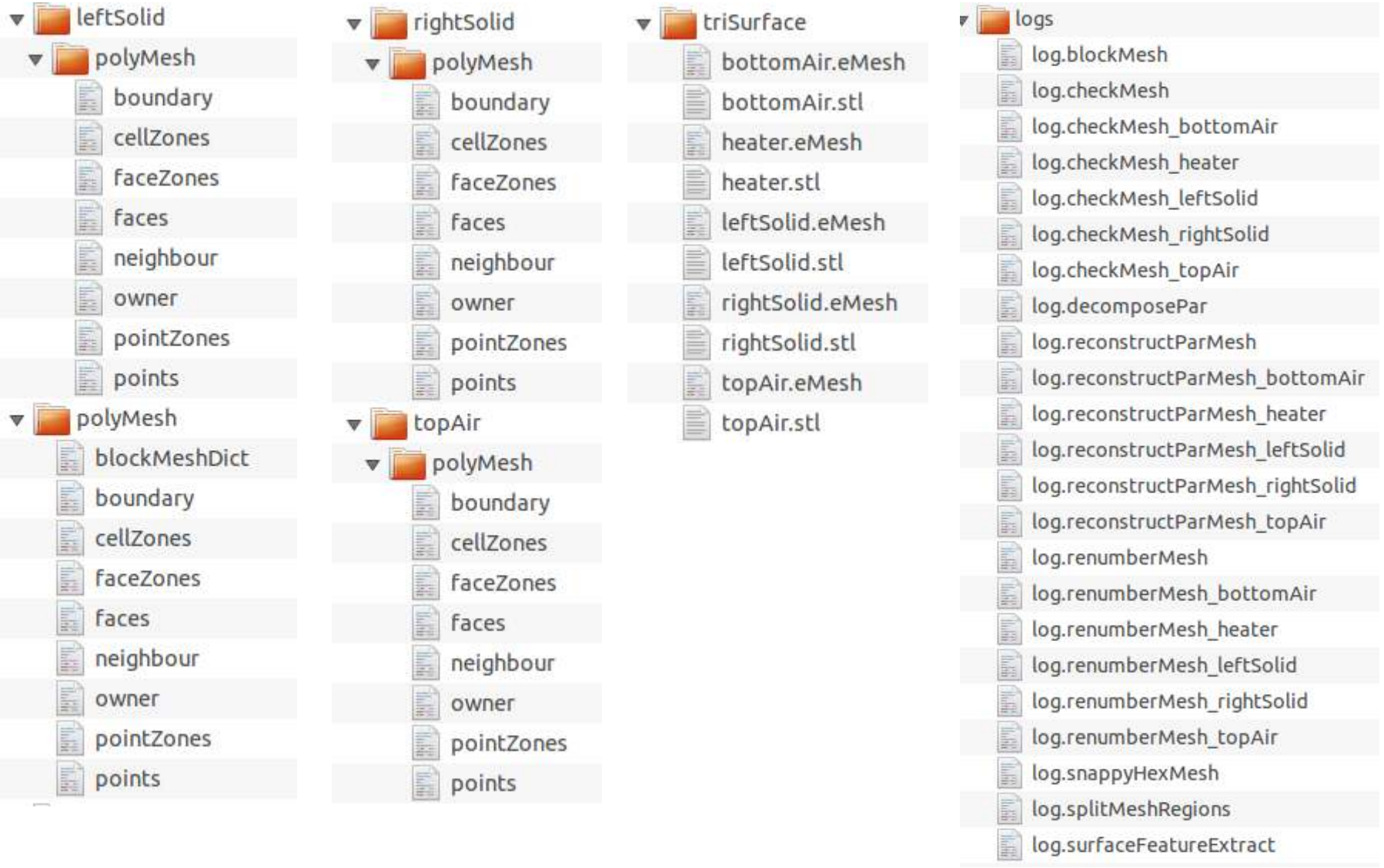
```
cp -r 3/polyMesh  constant
cp -r 3/$REGION_FLOW_1 constant
cp -r 3/$REGION_FLOW_2 constant
cp -r 3/$REGION_SOLID_1 constant
cp -r 3/$REGION_SOLID_2 constant
cp -r 3/$REGION_SOLID_3 constant
rm -r 2
rm -r 3
```

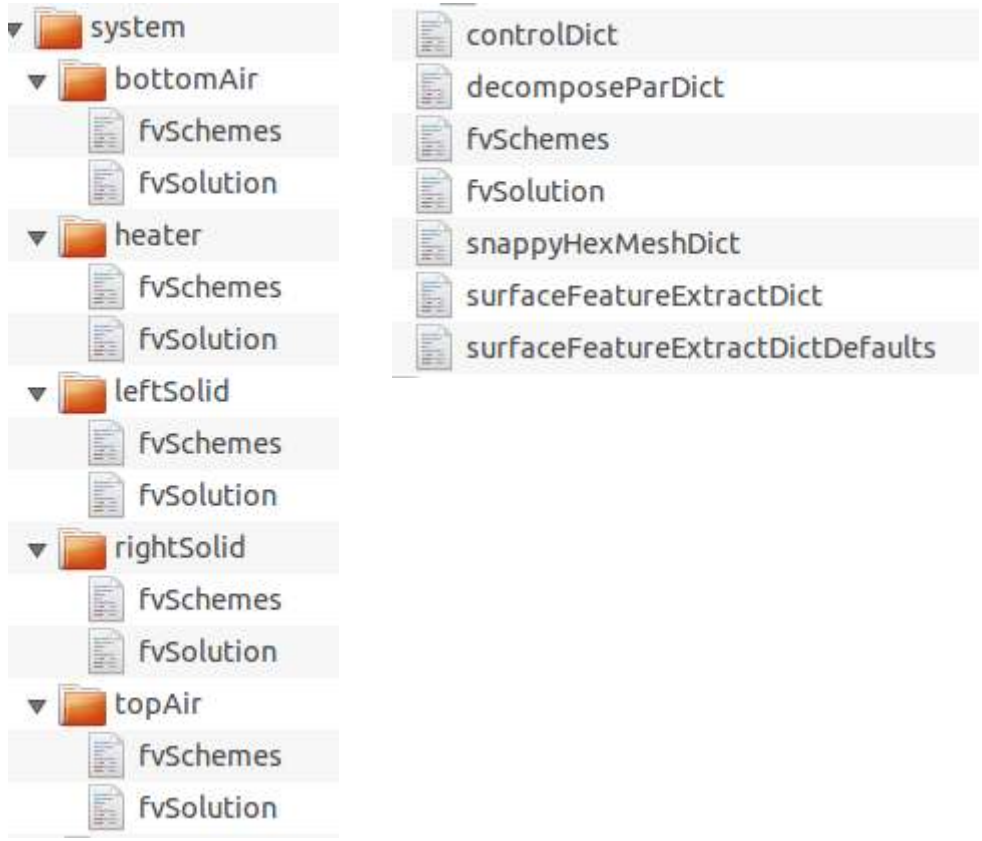
●メッシュ後のホルダ構成





かなりのファイルが作成される


leftSolid
polyMesh
boundary
cellZones
faceZones
faces
neighbour
owner
pointZones
points
polyMesh
blockMeshDict
boundary
cellZones
faceZones
faces
neighbour
owner
pointZones
points
rightSolid
polyMesh
boundary
cellZones
faceZones
faces
neighbour
owner
pointZones
points
topAir
polyMesh
boundary
cellZones
faceZones
faces
neighbour
owner
pointZones
points
triSurface
bottomAir.eMesh
bottomAir.stl
heater.eMesh
heater.stl
leftSolid.eMesh
leftSolid.stl
rightSolid.eMesh
rightSolid.stl
topAir.eMesh
topAir.stl
logs
log.blockMesh
log.checkMesh
log.checkMesh_bottomAir
log.checkMesh_heater
log.checkMesh_leftSolid
log.checkMesh_rightSolid
log.checkMesh_topAir
log.decomposePar
log.reconstructParMesh
log.reconstructParMesh_bottomAir
log.reconstructParMesh_heater
log.reconstructParMesh_leftSolid
log.reconstructParMesh_rightSolid
log.reconstructParMesh_topAir
log.renumberMesh
log.renumberMesh_bottomAir
log.renumberMesh_heater
log.renumberMesh_leftSolid
log.renumberMesh_rightSolid
log.renumberMesh_topAir
log.snappyHexMesh
log.splitMeshRegions
log.surfaceFeatureExtract

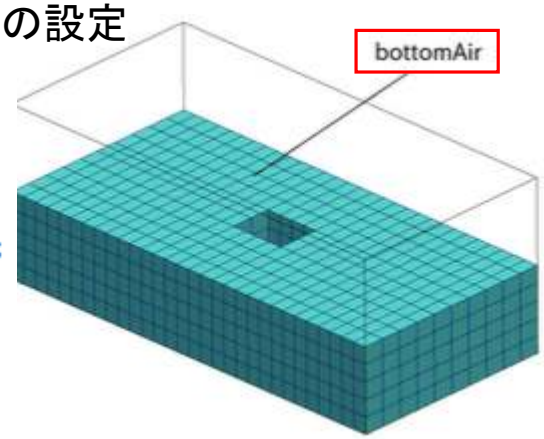

system
bottomAir
fvSchemes
fvSolution
heater
fvSchemes
fvSolution
leftSolid
fvSchemes
fvSolution
rightSolid
fvSchemes
fvSolution
topAir
fvSchemes
fvSolution
controlDict
decomposeParDict
fvSchemes
fvSolution
snappyHexMeshDict
surfaceFeatureExtractDict
surfaceFeatureExtractDictDefaults

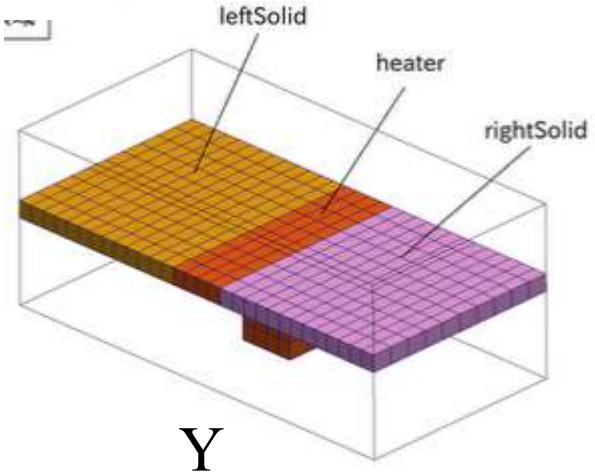
●境界設定　bottomAirのboundary

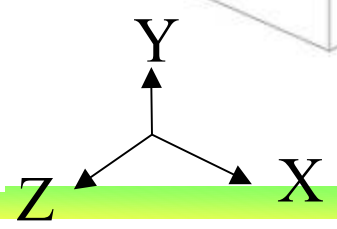
```
8
(
    minX
    {
        type            patch;
        nFaces          128;
        startFace       17022;
    }
    maxX
    {
        type            patch;
        nFaces          119;
        startFace       17150;
    }
    minY
    {
        type            wall;
        nFaces          1168;
        startFace       17269;
    }
    minZ
    {
        type            wall;
        nFaces          348;
        startFace       18437;
    }
    maxZ
    {
        type            wall;
        nFaces          339;
        startFace       18785;
    }
```

```
    bottomAir_to_rightSolid
    {
        type            mappedWall;
        nFaces          520;
        startFace       19124;
        sampleMode      nearestPatchFace;
        sampleRegion    rightSolid;
        samplePatch     rightSolid_to_bottomAir;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    bottomAir_to_heater
    {
        type            mappedWall;
        nFaces          368;
        startFace       19644;
        sampleMode      nearestPatchFace;
        sampleRegion    heater;
        samplePatch     heater_to_bottomAir;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    bottomAir_to_leftSolid
    {
        type            mappedWall;
        nFaces          520;
        startFace       20012;
        sampleMode      nearestPatchFace;
        sampleRegion    leftSolid;
        samplePatch     leftSolid_to_bottomAir;
        offsetMode      uniform;
        offset          (0 0 0);
    }
```

流体-固体に特有の設定



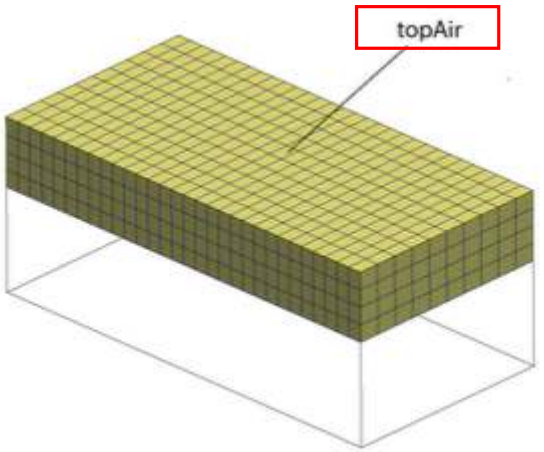


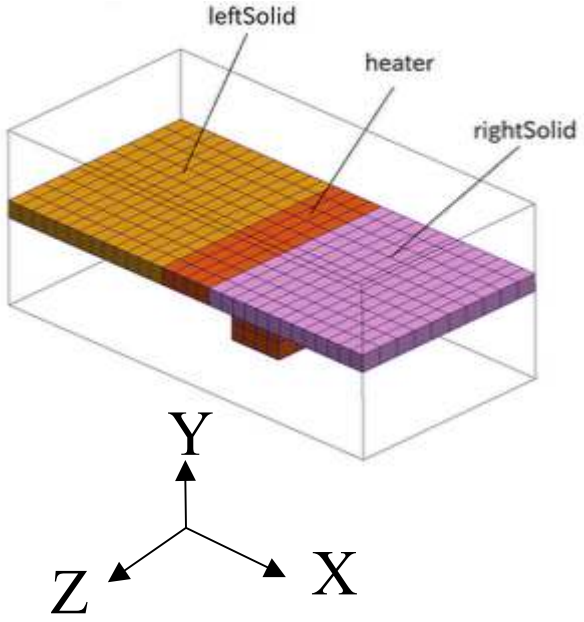


patch, wallの設定は後で修正する

●境界設定　topAirのboundary

```
8
(
    minX
    {
        type            patch;
        nFaces          112;
        startFace       15656;
    }
    maxX
    {
        type            patch;
        nFaces          112;
        startFace       15768;
    }
    maxY
    {
        type            wall;
        nFaces          1200;
        startFace       15880;
    }
    minZ
    {
        type            wall;
        nFaces          312;
        startFace       17080;
    }
    maxZ
    {
        type            wall;
        nFaces          312;
        startFace       17392;
    }
    topAir_to_rightSolid
    {
        type            mappedWall;
        nFaces          520;
        startFace       17704;
        sampleMode      nearestPatchFace;
        sampleRegion    rightSolid;
        samplePatch     rightSolid_to_topAir;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    topAir_to_leftSolid
    {
        type            mappedWall;
        nFaces          520;
        startFace       18224;
        sampleMode      nearestPatchFace;
        sampleRegion    leftSolid;
        samplePatch     leftSolid_to_topAir;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    topAir_to_heater
    {
        type            mappedWall;
        nFaces          160;
        startFace       18744;
        sampleMode      nearestPatchFace;
        sampleRegion    heater;
        samplePatch     heater_to_topAir;
        offsetMode      uniform;
        offset          (0 0 0);
    }
```





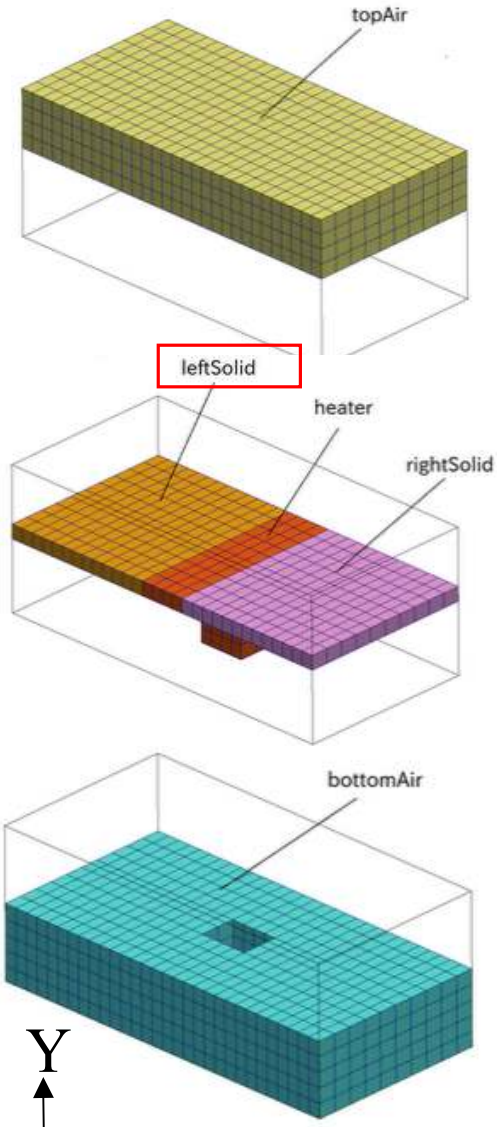
patch，wallの設定は後で修正する

●境界設定　leftSolidのboundary

```
6
(
    minX
    {
        type            patch;
        nFaces          40;
        startFace       2508;
    }
    minZ
    {
        type            wall;
        nFaces          52;
        startFace       2548;
    }
    maxZ
    {
        type            wall;
        nFaces          52;
        startFace       2600;
    }
```

```
    leftSolid_to_topAir
    {
        type            mappedWall;
        nFaces          520;
        startFace       2652;
        sampleMode      nearestPatchFace;
        sampleRegion    topAir;
        samplePatch     topAir_to_leftSolid;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    leftSolid_to_heater
    {
        type            mappedWall;
        nFaces          40;
        startFace       3172;
        sampleMode      nearestPatchFace;
        sampleRegion    heater;
        samplePatch     heater_to_leftSolid;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    leftSolid_to_bottomAir
    {
        type            mappedWall;
        nFaces          520;
        startFace       3212;
        sampleMode      nearestPatchFace;
        sampleRegion    bottomAir;
        samplePatch     bottomAir_to_leftSolid;
        offsetMode      uniform;
        offset          (0 0 0);
    }
```

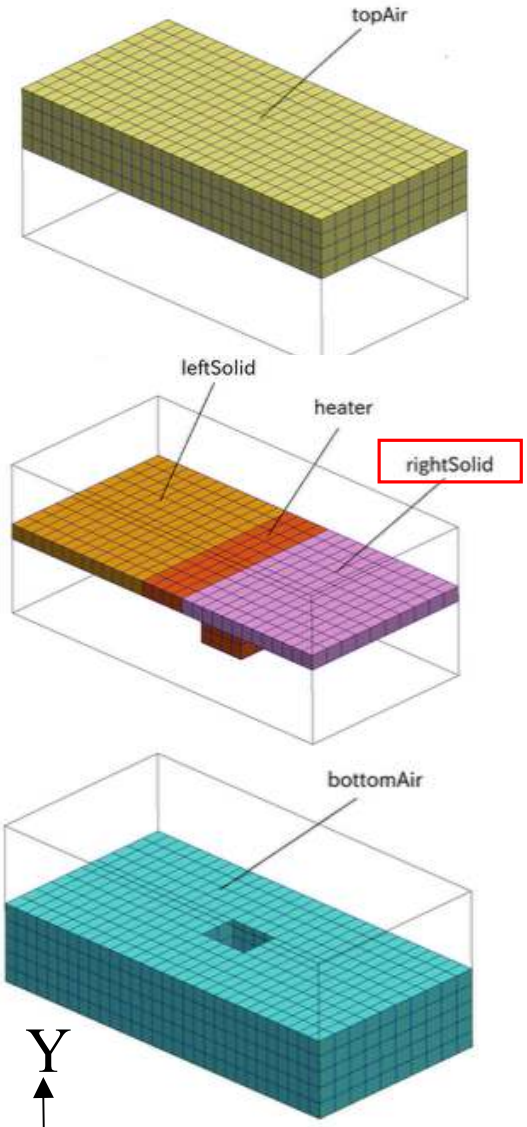
patch，wallの設定は後で修正する



●境界設定　rightSolidのboundary

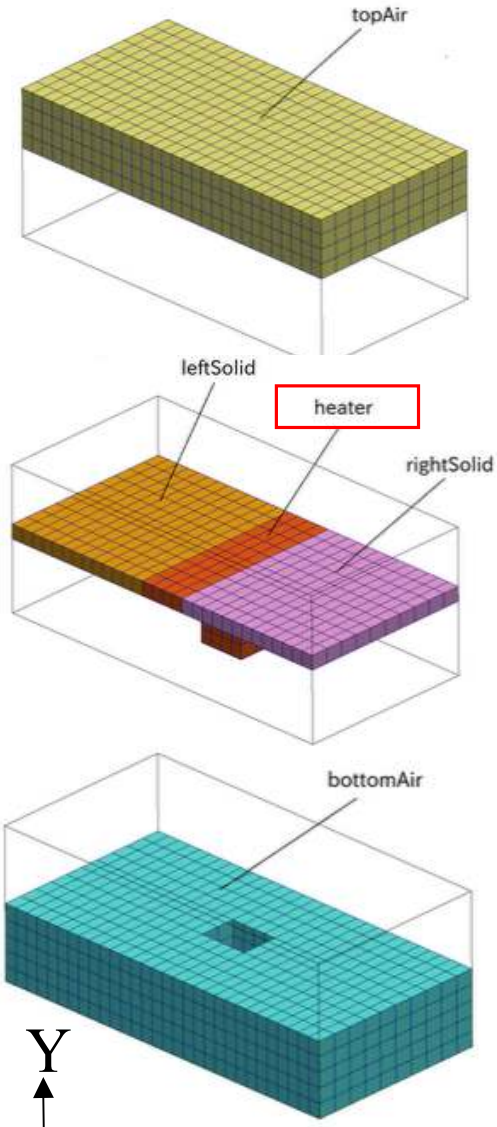
```
6
(
    maxX
    {
        type            patch;
        nFaces          40;
        startFace       2508;
    }
    minZ
    {
        type            wall;
        nFaces          52;
        startFace       2548;
    }
    maxZ
    {
        type            wall;
        nFaces          52;
        startFace       2600;
    }
    rightSolid_to_bottomAir
    {
        type            mappedWall;
        nFaces          520;
        startFace       2652;
        sampleMode      nearestPatchFace;
        sampleRegion    bottomAir;
        samplePatch     bottomAir_to_rightSolid;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    rightSolid_to_topAir
    {
        type            mappedWall;
        nFaces          520;
        startFace       3172;
        sampleMode      nearestPatchFace;
        sampleRegion    topAir;
        samplePatch     topAir_to_rightSolid;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    rightSolid_to_heater
    {
        type            mappedWall;
        nFaces          40;
        startFace       3692;
        sampleMode      nearestPatchFace;
        sampleRegion    heater;
        samplePatch     heater_to_rightSolid;
        offsetMode      uniform;
        offset          (0 0 0);
    }
```

patch，wallの設定は後で修正する

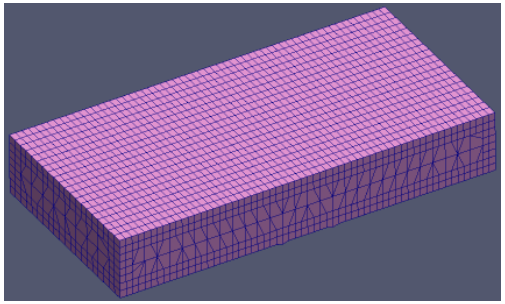


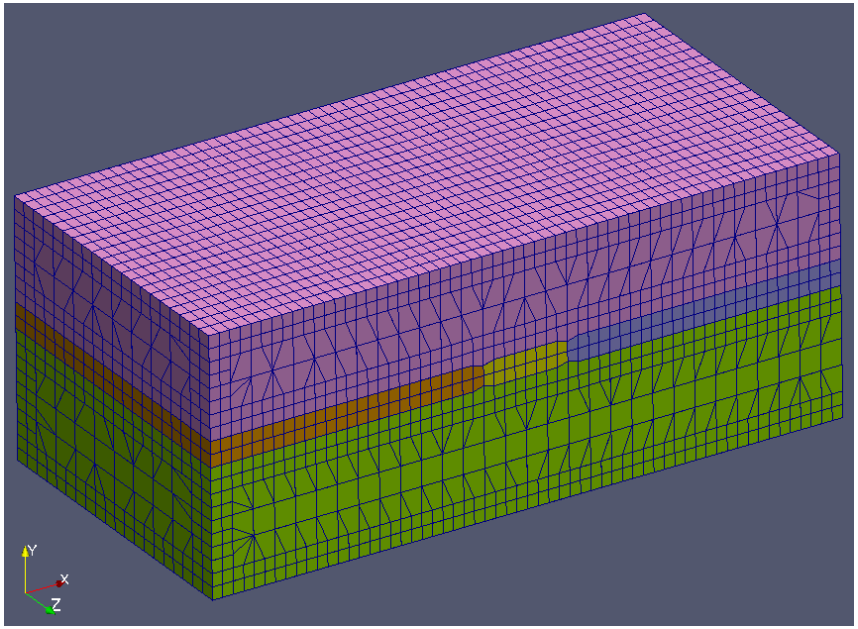
●境界設定　heaterのboundary

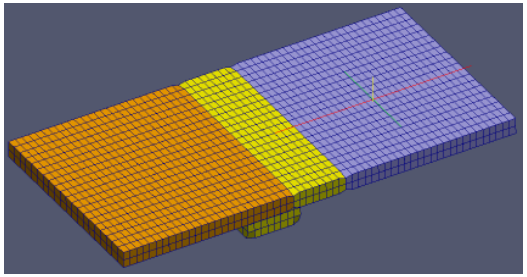
```
7
(
    minY
    {
        type            wall;
        nFaces          32;
        startFace       1584;
    }
    minZ
    {
        type            wall;
        nFaces          16;
        startFace       1616;
    }
    maxZ
    {
        type            wall;
        nFaces          16;
        startFace       1632;
    }
    heater_to_bottomAir
    {
        type            mappedWall;
        nFaces          368;
        startFace       1648;
        sampleMode      nearestPatchFace;
        sampleRegion    bottomAir;
        samplePatch     bottomAir_to_heater
        offsetMode      uniform;
        offset          (0 0 0);
    }
    heater_to_leftSolid
    {
        type            mappedWall;
        nFaces          40;
        startFace       2016;
        sampleMode      nearestPatchFace;
        sampleRegion    leftSolid;
        samplePatch     leftSolid_to_heater;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    heater_to_rightSolid
    {
        type            mappedWall;
        nFaces          40;
        startFace       2056;
        sampleMode      nearestPatchFace;
        sampleRegion    rightSolid;
        samplePatch     rightSolid_to_heater;
        offsetMode      uniform;
        offset          (0 0 0);
    }
    heater_to_topAir
    {
        type            mappedWall;
        nFaces          160;
        startFace       2096;
        sampleMode      nearestPatchFace;
        sampleRegion    topAir;
        samplePatch     topAir_to_heater;
        offsetMode      uniform;
        offset          (0 0 0);
    }
```



●メッシュ状態の確認

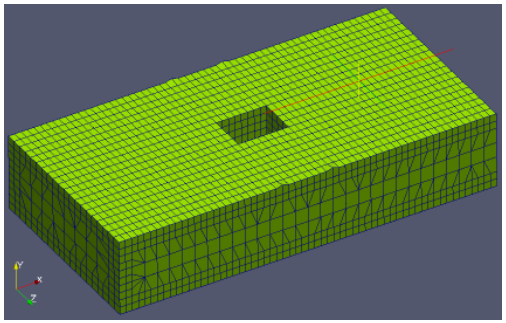


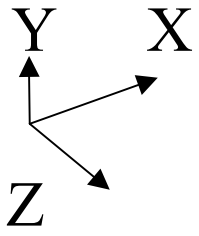


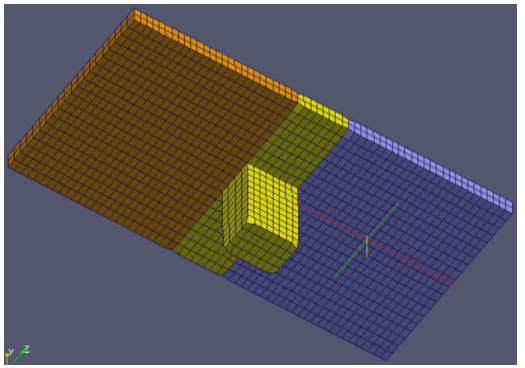


流体部分

固体部分







●コマンド，pyFoamを利用した境界設定の作成

先ほどのホルダにT,U,p,p_rgh,rho,alphat
のダミーファイルを設定する

T

```
FoamFile
{
 version 2.0;
 format ascii;
 class volScalarField;
 location "0";
 object T;
}

dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{

}       // ********************
```

U

```
FoamFile
{
 version 2.0;
 format ascii;
 class volVectorField;
 location "0";
 object U;
}

dimensions [ 0 1 -1 0 0 0 0 ];

internalField uniform (0.01 0 0);

boundaryField
{

}       // ************************
```

ダミーファイルの状態

alphat

```
FoamFile
{
 version 2.0;
 format ascii;
 class volScalarField;
 location "0";
 object alphat;
}

dimensions [ 1 -1 -1 0 0 0 0 ];

internalField uniform 0;

boundaryField
{

}       // ***********************
```

p

```
FoamFile
{
 version 2.0;
 format ascii;
 class volScalarField;
 location "0";
 object p;
}

dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{

}       // ***********************
```

p_rgh

```
FoamFile
{
 version 2.0;
 format ascii;
 class volScalarField;
 location "0";
 object p_rgh;
}

dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{

}       // ***********************
```

ダミーファイルの状態

rho

```
FoamFile
{
 version 2.0;
 format ascii;
 class volScalarField;
 location "0";
 object rho;
}

dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{

}       // ********************
```

●コマンドによるメッシュ作成手順

①ファイルの設定

```
REGION_FLOW_1=bottomAir
REGION_FLOW_2=topAir
REGION_SOLID_1=heater
REGION_SOLID_2=leftSolid
REGION_SOLID_3=rightSolid
```

流体，固体の各領域名の設定
constant，systemのホルダ設定に利用

②流体部分の設定

```
for i in $REGION_FLOW_1 $REGION_FLOW_2
   do
      rm -f -r temp_set_boundary
      mkdir temp_set_boundary
      mkdir temp_set_boundary/0
      mkdir temp_set_boundary/constant
      mkdir temp_set_boundary/system
```

２領域

利用するホルダの準備

```
cp -r case_set/case1/set/backup/initial_set_file/T       temp_set_boundary/0
cp -r case_set/case1/set/backup/initial_set_file/U       temp_set_boundary/0
cp -r case_set/case1/set/backup/initial_set_file/alphat  temp_set_boundary/0
cp -r case_set/case1/set/backup/initial_set_file/p       temp_set_boundary/0
cp -r case_set/case1/set/backup/initial_set_file/p_rgh   temp_set_boundary/0
cp -r case_set/case1/set/backup/initial_set_file/rho     temp_set_boundary/0
cp -r constant/$i/polyMesh                               temp_set_boundary/constant
cp -r case_set/case1/set/controlDict_case1_mesh          temp_set_boundary/system/controlDict
```

利用するファイルの準備

②流体部分の設定

```
        cd temp_set_boundary
        $runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/T
        $runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/U
        $runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/alphat
        $runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/p
        $runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/p_rgh
        $runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/rho
        cd ..

        rm -f -r 0/$i
        mkdir 0/$i
        cp -r temp_set_boundary/0/* 0/$i
done
```

pyFoamを利用して境界を設定

設定したファイルを正規の位置に設定

③固体部分の設定

```
for i in $REGION_SOLID_1 $REGION_SOLID_2 $REGION_SOLID_3
   do
      rm -f -r temp_set_boundary
      mkdir temp_set_boundary
      mkdir temp_set_boundary/0
      mkdir temp_set_boundary/constant
      mkdir temp_set_boundary/system

      cp -r case_set/case1/set/backup/initial_set_file/T      temp_set_boundary/0
      cp -r case_set/case1/set/backup/initial_set_file/p      temp_set_boundary/0
      cp -r case_set/case1/set/backup/initial_set_file/rho    temp_set_boundary/0
      cp -r constant/$i/polyMesh                              temp_set_boundary/constant
      cp -r case_set/case1/set/controlDict_case1_mesh         temp_set_boundary/system/controlDict

      cd temp_set_boundary
      $runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/T
      $runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/p
      $runApplication pyFoamCreateBoundaryPatches.py --clear-unused 0/rho
      cd ..

      rm -f -r 0/$i
      mkdir 0/$i
      cp -r temp_set_boundary/0/* 0/$i
done
```

3領域

利用するホルダの準備

利用するファイルの準備

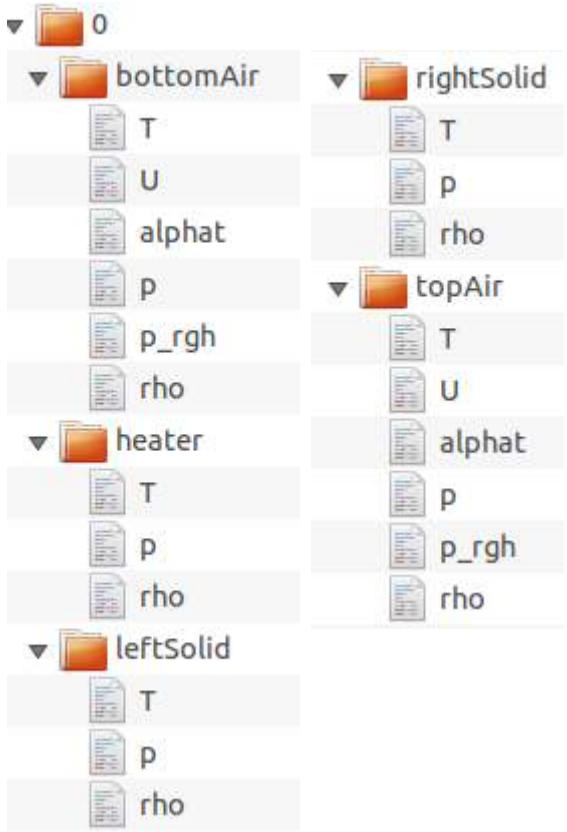
pyFoamを利用して境界を設定

設定したファイルを正規の位置に設定

④不要ホルダの削除

```
rm -f -r temp_set_boundary
```

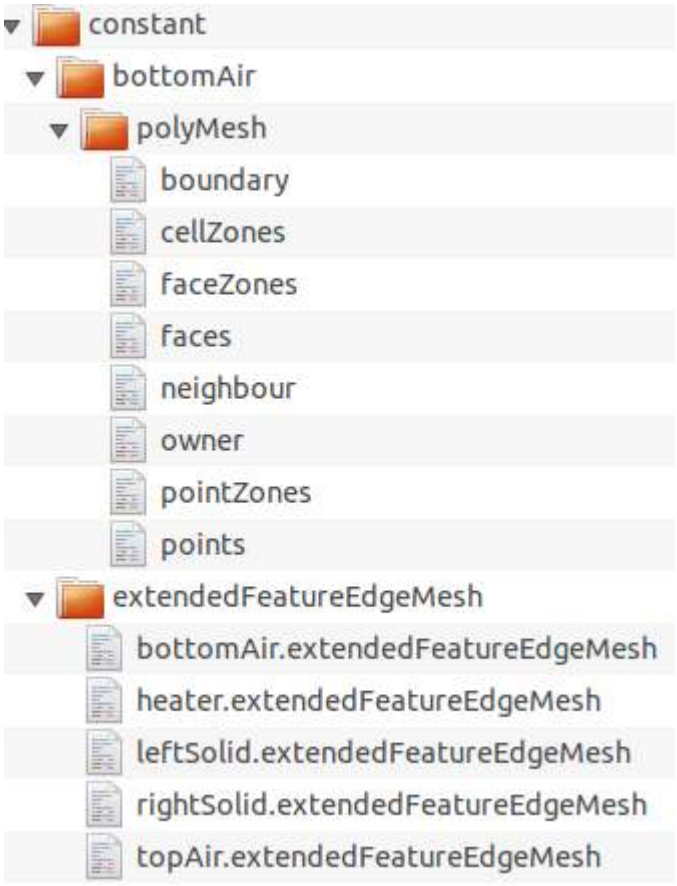
●境界設定後の0ホルダの状態

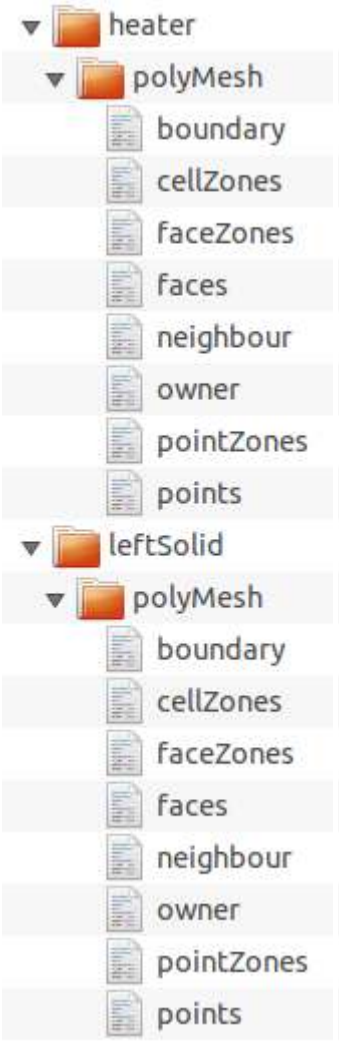


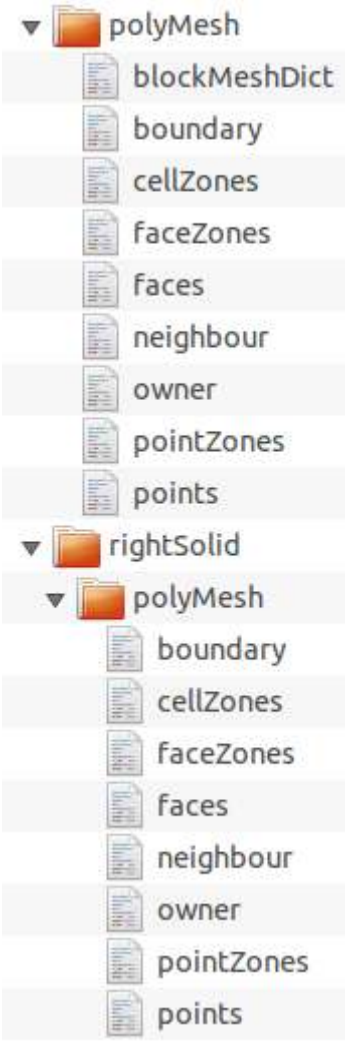
0ホルダの各領域には,計算に必要なファイルが用意される。
ただし，この時点では領域で設定される値はダミーデータとなっている。

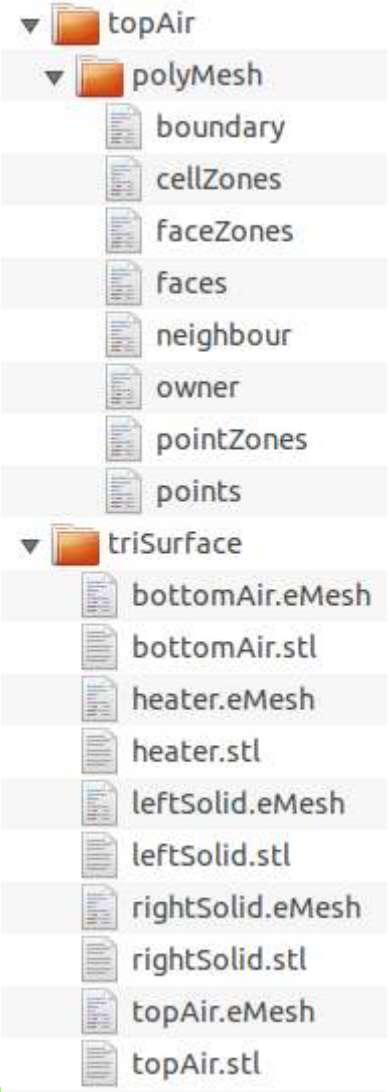
●その他のホルダの状態

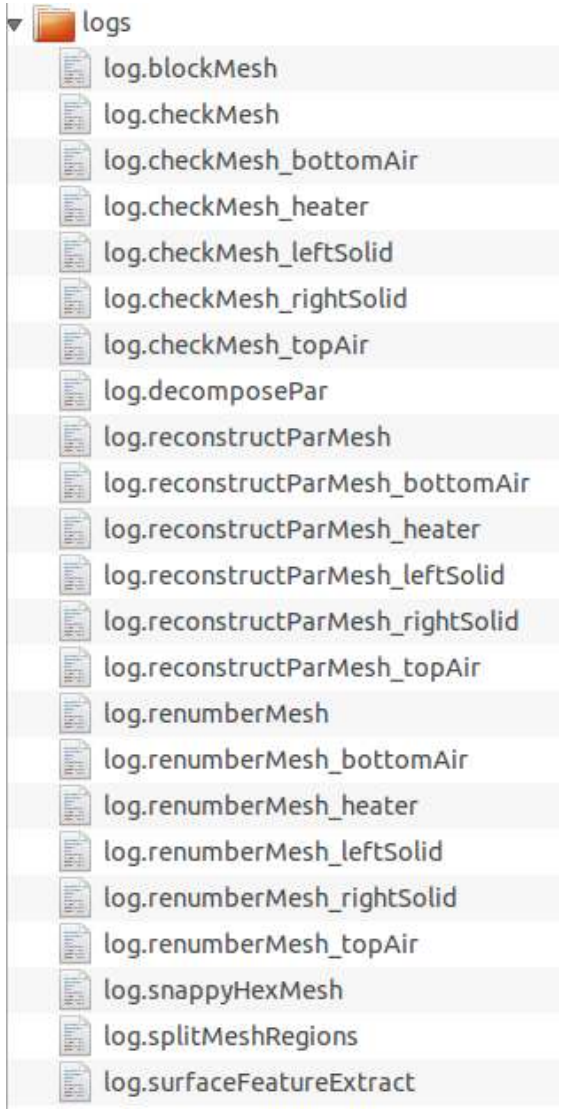
constant以下のホルダ



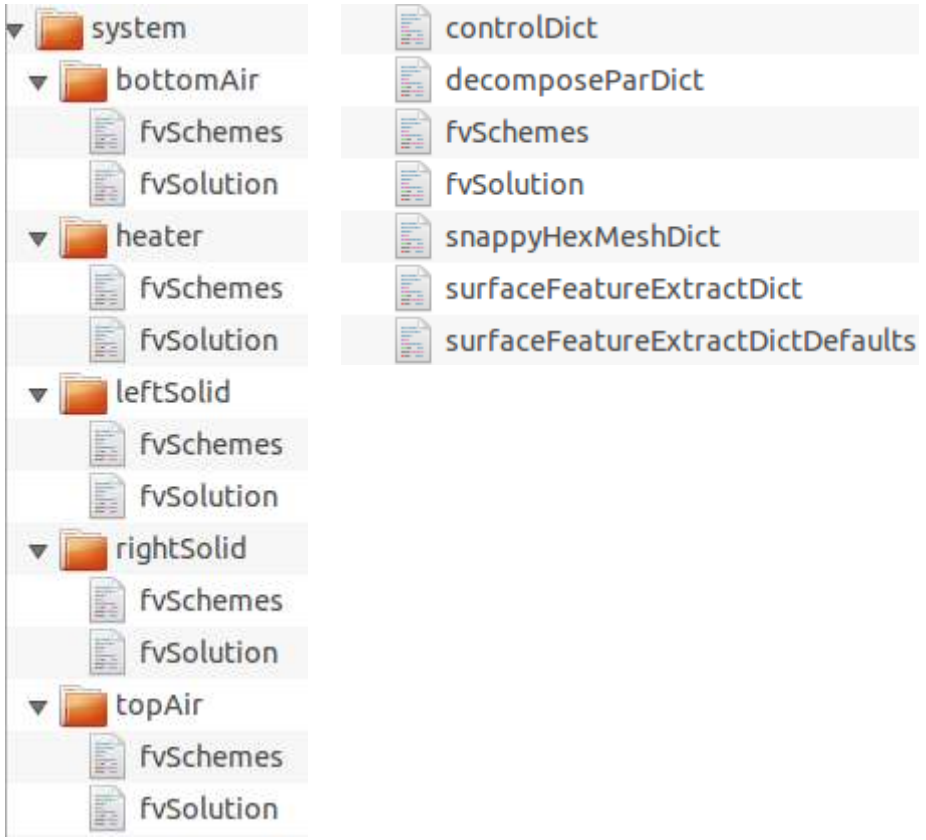








system以下のホルダ



● 0にある初期値とconstantにある各領域のboundaryを手修正

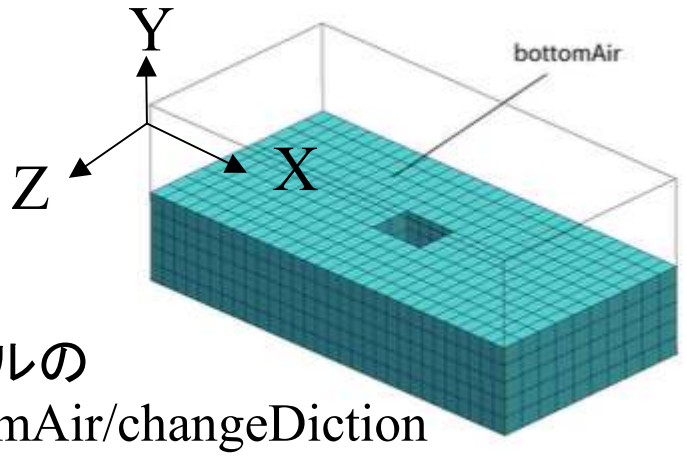
修正は，チュートリアルのsystemホルダの各領域ホルダにあるchangeDictionaryDictを参照しながら，面倒であるが手で修正する。

changeDictionaryDictを作成して一括に変更する方法もあるが，今回は手動で全部実施している。このため，手間がかかる。

● 0/bottomAir/Tの修正

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  minY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  bottomAir_to_rightSolid
  {
    type mappedWall;
  }
  bottomAir_to_heater
  {
    type mappedWall;
  }
  bottomAir_to_leftSolid
  {
    type mappedWall;
  }
}       // ******************
```

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  minX
  {
    type zeroGradient;
  }

  maxX
  {
    type zeroGradient;
  }

  minY
  {
    type zeroGradient;
  }

  minZ
  {
    type zeroGradient;
  }

  maxZ
  {
    type zeroGradient;
  }
```



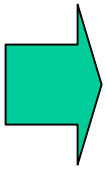
チュートリアルの
system/bottomAir/changeDictionayDictを参照にして設定する

```
bottomAir_to_rightSolid
{
      type            compressible::turbulentTemperatureCoupledBaffleMixed;
      value           uniform 300;
      neighbourFieldName T;
      kappa           fluidThermo;
      kappaName       none;
}

bottomAir_to_heater
{
      type            compressible::turbulentTemperatureCoupledBaffleMixed;
      value           uniform 300;
      neighbourFieldName T;
      kappa           fluidThermo;
      kappaName       none;
}

bottomAir_to_leftSolid
{
      type            compressible::turbulentTemperatureCoupledBaffleMixed;
      value           uniform 300;
      neighbourFieldName T;
      kappa           fluidThermo;
      kappaName       none;
```

面倒であるが手で修正する（以降同じ）

● 0/bottomAir/Uの修正

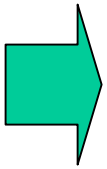
```
dimensions [ 0 1 -1 0 0 0 0 ];

internalField uniform (0.01 0 0);

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  minY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  bottomAir_to_rightSolid
  {
    type mappedWall;
  }
  bottomAir_to_heater
  {
    type mappedWall;
  }
  bottomAir_to_leftSolid
  {
    type mappedWall;
  }
}       // ************************
```



```
dimensions [ 0 1 -1 0 0 0 0 ];

internalField uniform (0.01 0 0);

boundaryField
{
  minX
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  maxX
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  minY
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  minZ
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  maxZ
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  bottomAir_to_rightSolid
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  bottomAir_to_heater
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  bottomAir_to_leftSolid
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }
}       // ************************
```

● 0/bottomAir/alphatの修正

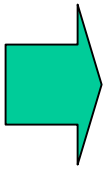
```
dimensions [ 1 -1 -1 0 0 0 0 ];

internalField uniform 0;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  minY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  bottomAir_to_rightSolid
  {
    type mappedWall;
  }
  bottomAir_to_heater
  {
    type mappedWall;
  }
  bottomAir_to_leftSolid
  {
    type mappedWall;
  }
}       // ********************
```



```
dimensions [ 1 -1 -1 0 0 0 0 ];

internalField uniform 0;

boundaryField
{
  minX
  {
        type            calculated;
        value           uniform 0;
  }

  maxX
  {
        type            calculated;
        value           uniform 0;
  }

  minY
  {
        type            calculated;
        value           uniform 0;
  }

  minZ
  {
        type            calculated;
        value           uniform 0;
  }

  maxZ
  {
        type            calculated;
        value           uniform 0;
  }

  bottomAir_to_rightSolid
  {
        type            calculated;
        value           uniform 0;
  }

  bottomAir_to_rightSolid
  {
        type            calculated;
        value           uniform 0;
  }

  bottomAir_to_heater
  {
        type            calculated;
        value           uniform 0;
  }

  bottomAir_to_leftSolid
  {
        type            calculated;
        value           uniform 0;
  }
}       // ************************
```

● 0/bottomAir/pの修正

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  minY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  bottomAir_to_rightSolid
  {
    type mappedWall;
  }
  bottomAir_to_heater
  {
    type mappedWall;
  }
  bottomAir_to_leftSolid
  {
    type mappedWall;
  }
}       // ********************
```



```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
        type            calculated;
        value           uniform 100000;
  }

  maxX
  {
        type            calculated;
        value           uniform 100000;
  }

  minY
  {
        type            calculated;
        value           uniform 100000;
  }

  minZ
  {
        type            calculated;
        value           uniform 100000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 100000;
  }

  bottomAir_to_rightSolid
  {
        type            calculated;
        value           uniform 100000;
  }

  bottomAir_to_heater
  {
        type            calculated;
        value           uniform 100000;
  }

  bottomAir_to_leftSolid
  {
        type            calculated;
        value           uniform 100000;
  }
}       // ********************
```

● 0/bottomAir/p_rghの修正

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  minY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  bottomAir_to_rightSolid
  {
    type mappedWall;
  }
  bottomAir_to_heater
  {
    type mappedWall;
  }
  bottomAir_to_leftSolid
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  maxX
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  minY
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  minZ
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  maxZ
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }
  bottomAir_to_rightSolid
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  bottomAir_to_heater
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  bottomAir_to_leftSolid
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }
}       // ********************************
```

● 0/bottomAir/rhoの修正

```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  minY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  bottomAir_to_rightSolid
  {
    type mappedWall;
  }
  bottomAir_to_heater
  {
    type mappedWall;
  }
  bottomAir_to_leftSolid
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  minX
  {
        type            calculated;
        value           uniform 8000;
  }

  maxX
  {
        type            calculated;
        value           uniform 8000;
  }

  minY
  {
        type            calculated;
        value           uniform 8000;
  }

  minZ
  {
        type            calculated;
        value           uniform 8000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 8000;
  }
```

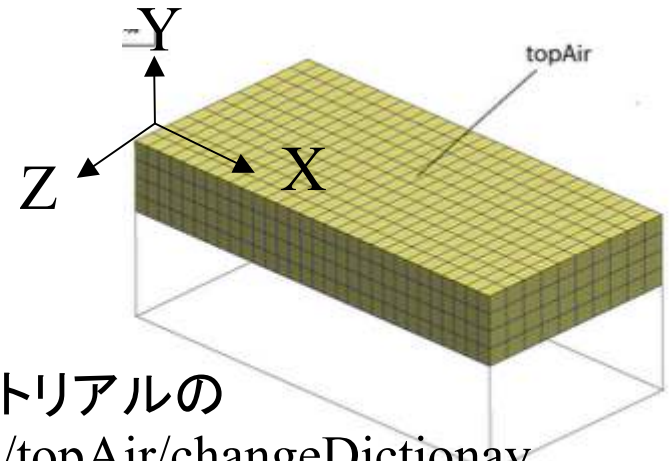
```
  bottomAir_to_rightSolid
  {
        type            calculated;
        value           uniform 0;
  }

  bottomAir_to_heater
  {
        type            calculated;
        value           uniform 0;
  }

  bottomAir_to_leftSolid
  {
        type            calculated;
        value           uniform 0;
  }
}       // ************************
```

● 0/topAir/Tの修正

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  maxY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  topAir_to_rightSolid
  {
    type mappedWall;
  }
  topAir_to_leftSolid
  {
    type mappedWall;
  }
  topAir_to_heater
  {
    type mappedWall;
  }
}       // ******************
```

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  minX
  {
        type            fixedValue;
        value           uniform 300;
  }

  maxX
  {
        type            inletOutlet;
        value           uniform 300;
        inletValue      uniform 300;
  }

  maxY
  {
    type zeroGradient;
  }

  minZ
  {
    type zeroGradient;
  }

  maxZ
  {
    type zeroGradient;
  }
```

```
  topAir_to_rightSolid
  {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           fluidThermo;
        kappaName       none;
  }

  topAir_to_leftSolid
  {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           fluidThermo;
        kappaName       none;
  }

  topAir_to_heater
  {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           fluidThermo;
        kappaName       none;
  }
}       // ************************************************************************
```



チュートリアルの
system/topAir/changeDictionay
Dictを参照にして設定する

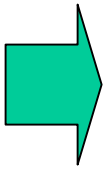
● 0/topAir/Uの修正

```
dimensions [ 0 1 -1 0 0 0 0 ];

internalField uniform (0.01 0 0);

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  maxY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  topAir_to_rightSolid
  {
    type mappedWall;
  }
  topAir_to_leftSolid
  {
    type mappedWall;
  }
  topAir_to_heater
  {
    type mappedWall;
  }
}       // ************************
```

```
dimensions [ 0 1 -1 0 0 0 0 ];

internalField uniform (0.1 0 0);

boundaryField
{
  minX
  {
        type            fixedValue;
        value           uniform ( 0.1 0 0 );
  }

  maxX
  {
        type            inletOutlet;
        value           uniform ( 0.1 0 0 );
        inletValue      uniform ( 0 0 0 );
  }

  maxY
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  minZ
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  maxZ
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  topAir_to_rightSolid
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  topAir_to_leftSolid
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }

  topAir_to_heater
  {
        type            fixedValue;
        value           uniform ( 0 0 0 );
  }
}       // ********************************
```

● O/topAir/alphatの修正

```
dimensions [ 1 -1 -1 0 0 0 0 ];

internalField uniform 0;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  maxY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  topAir_to_rightSolid
  {
    type mappedWall;
  }
  topAir_to_leftSolid
  {
    type mappedWall;
  }
  topAir_to_heater
  {
    type mappedWall;
  }
}       // ********************
```



```
dimensions [ 1 -1 -1 0 0 0 0 ];

internalField uniform 0;

boundaryField
{
  minX
  {
        type            calculated;
        value           uniform 0;
  }

  maxX
  {
        type            calculated;
        value           uniform 0;
  }

  maxY
  {
        type            calculated;
        value           uniform 0;
  }

  minZ
  {
        type            calculated;
        value           uniform 0;
  }

  maxZ
  {
        type            calculated;
        value           uniform 0;
  }

  topAir_to_rightSolid
  {
        type            calculated;
        value           uniform 0;
  }

  topAir_to_leftSolid
  {
        type            calculated;
        value           uniform 0;
  }

  topAir_to_heater
  {
        type            calculated;
        value           uniform 0;
  }
}       // ************************
```

● 0/topAir/pの修正

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  maxY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  topAir_to_rightSolid
  {
    type mappedWall;
  }
  topAir_to_leftSolid
  {
    type mappedWall;
  }
  topAir_to_heater
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
        type            calculated;
        value           uniform 100000;
  }

  maxX
  {
        type            calculated;
        value           uniform 100000;
  }

  maxY
  {
        type            calculated;
        value           uniform 100000;
  }
  minZ
  {
        type            calculated;
        value           uniform 100000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 100000;
  }
  topAir_to_rightSolid
  {
        type            calculated;
        value           uniform 100000;
  }

  topAir_to_leftSolid
  {
        type            calculated;
        value           uniform 100000;
  }

  topAir_to_heater
  {
        type            calculated;
        value           uniform 100000;
  }
}       // ******************************
```

● 0/topAir/p_rghの修正

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  maxY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  topAir_to_rightSolid
  {
    type mappedWall;
  }
  topAir_to_leftSolid
  {
    type mappedWall;
  }
  topAir_to_heater
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  maxX
  {
        type            fixedValue;
        value           uniform 100000;
  }

  maxY
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  minZ
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  maxZ
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  topAir_to_rightSolid
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  topAir_to_leftSolid
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }

  topAir_to_heater
  {
        type            fixedFluxPressure;
        value           uniform 100000;
  }
}       // ********************************
```

● 0/topAir/rhoの修正

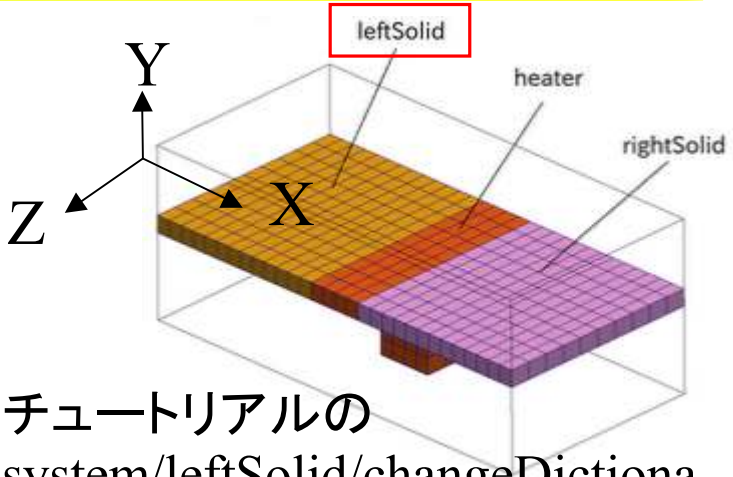
```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  maxX
  {
    type zeroGradient;
  }
  maxY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  topAir_to_rightSolid
  {
    type mappedWall;
  }
  topAir_to_leftSolid
  {
    type mappedWall;
  }
  topAir_to_heater
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  minX
  {
        type            calculated;
        value           uniform 8000;
  }

  maxX
  {
        type            calculated;
        value           uniform 8000;
  }

  maxY
  {
        type            calculated;
        value           uniform 8000;
  }

  minZ
  {
        type            calculated;
        value           uniform 8000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 8000;
  }

  topAir_to_rightSolid
  {
        type            calculated;
        value           uniform 8000;
  }

  topAir_to_leftSolid
  {
        type            calculated;
        value           uniform 8000;
  }

  topAir_to_heater
  {
        type            calculated;
        value           uniform 8000;
  }
}       // ****************************
```

● 0/leftSolid/Tの修正

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  leftSolid_to_topAir
  {
    type mappedWall;
  }
  leftSolid_to_heater
  {
    type mappedWall;
  }
  leftSolid_to_bottomAir
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  minX
  {
    type zeroGradient;
  }

  minZ
  {
    type zeroGradient;
  }

  maxZ
  {
    type zeroGradient;
  }
```



チュートリアルの
system/leftSolid/changeDictionaryDictを参照にして設定する

```
  leftSolid_to_topAir
  {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
  }

  leftSolid_to_heater
  {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
  }

  leftSolid_to_bottomAir
  {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
  }
}       // ************************************************************************
```

● 0/leftSolid/pの修正

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  leftSolid_to_topAir
  {
    type mappedWall;
  }
  leftSolid_to_heater
  {
    type mappedWall;
  }
  leftSolid_to_bottomAir
  {
    type mappedWall;
  }
}       // **********************
```

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minX
  {
        type            calculated;
        value           uniform 100000;
  }

  minZ
  {
        type            calculated;
        value           uniform 100000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 100000;
  }
```

```
  leftSolid_to_topAir
  {
        type            calculated;
        value           uniform 0;
  }

  leftSolid_to_heater
  {
        type            calculated;
        value           uniform 0;
  }

  leftSolid_to_bottomAir
  {
        type            calculated;
        value           uniform 0;
  }
}       // **************************
```

● 0/leftSolid/rhoの修正

```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  minX
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  leftSolid_to_topAir
  {
    type mappedWall;
  }
  leftSolid_to_heater
  {
    type mappedWall;
  }
  leftSolid_to_bottomAir
  {
    type mappedWall;
  }
}       // ********************
```

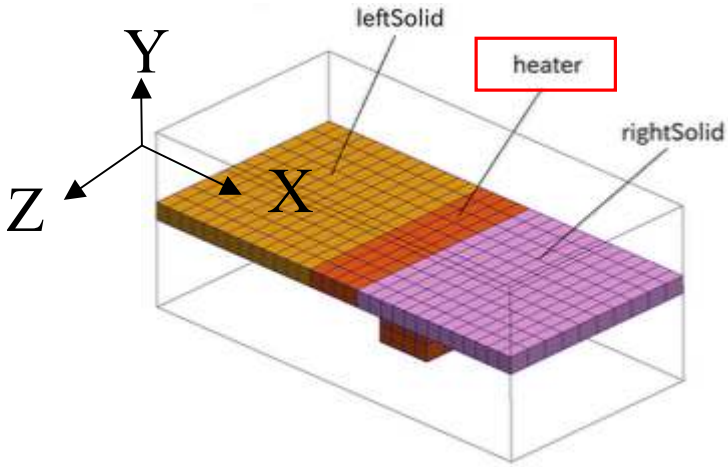
```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  minX
  {
        type            calculated;
        value           uniform 8000;
  }

  minZ
  {
        type            calculated;
        value           uniform 8000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 8000;
  }
```

```
  leftSolid_to_topAir
  {
        type            calculated;
        value           uniform 0;
  }

  leftSolid_to_heater
  {
        type            calculated;
        value           uniform 0;
  }

  leftSolid_to_bottomAir
  {
        type            calculated;
        value           uniform 0;
  }
}       // ************************
```

● O/heater/Tの修正

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  minY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  heater_to_bottomAir
  {
    type mappedWall;
  }
  heater_to_leftSolid
  {
    type mappedWall;
  }
  heater_to_rightSolid
  {
    type mappedWall;
  }
  heater_to_topAir
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  minY
  {
        type            fixedValue;
        value           uniform 500;
  }

  minZ
  {
        type            zeroGradient;
  }

  maxZ
  {
        type            zeroGradient;
  }

  heater_to_bottomAir
  {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
  }

  heater_to_leftSolid
  {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
  }
```



チュートリアルの
system/heater/changeDictionayDictを参照にして設定する

● O/heater/Tの修正

```
    heater_to_rightSolid
    {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
    }

    heater_to_topAir
    {
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
    }
}       // ************************************************************************
```

● O/heater/pの修正

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  heater_to_bottomAir
  {
    type mappedWall;
  }
  heater_to_leftSolid
  {
    type mappedWall;
  }
  heater_to_rightSolid
  {
    type mappedWall;
  }
  heater_to_topAir
  {
    type mappedWall;
  }
}       // **********************
```

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  minY
  {
        type            calculated;
        value           uniform 100000;
  }

  minZ
  {
        type            calculated;
        value           uniform 100000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 100000;
  }
```

```
  heater_to_bottomAir
  {
        type            calculated;
        value           uniform 0;
  }

  heater_to_leftSolid
  {
        type            calculated;
        value           uniform 0;
  }

  heater_to_rightSolid
  {
        type            calculated;
        value           uniform 0;
  }

  heater_to_topAir
  {
        type            calculated;
        value           uniform 0;
  }
}       // **********************
```

● O/heater/rhoの修正

```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  minY
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  heater_to_bottomAir
  {
    type mappedWall;
  }
  heater_to_leftSolid
  {
    type mappedWall;
  }
  heater_to_rightSolid
  {
    type mappedWall;
  }
  heater_to_topAir
  {
    type mappedWall;
  }
}       // ********************
```

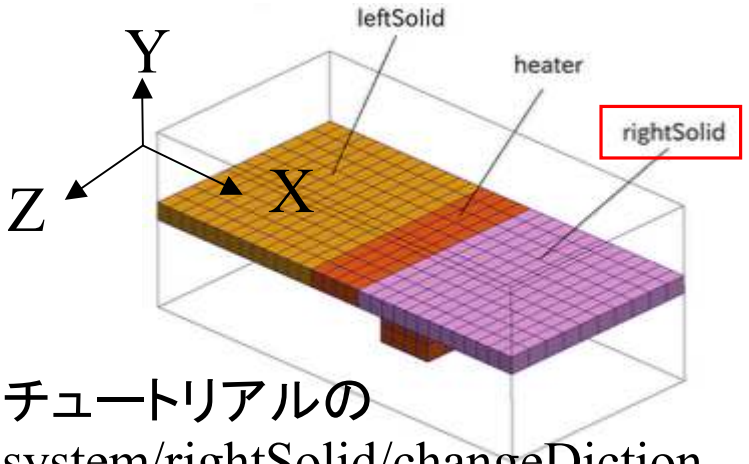
```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  minY
  {
        type            calculated;
        value           uniform 8000;
  }

  minZ
  {
        type            calculated;
        value           uniform 8000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 8000;
  }
```

```
  heater_to_bottomAir
  {
        type            calculated;
        value           uniform 0;
  }

  heater_to_leftSolid
  {
        type            calculated;
        value           uniform 0;
  }

  heater_to_rightSolid
  {
        type            calculated;
        value           uniform 0;
  }

  heater_to_topAir
  {
        type            calculated;
        value           uniform 0;
  }
}       // ************************
```

● O/rightSolid/Tの修正

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  maxX
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  rightSolid_to_bottomAir
  {
    type mappedWall;
  }
  rightSolid_to_topAir
  {
    type mappedWall;
  }
  rightSolid_to_heater
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 0 0 0 1 0 0 0 ];

internalField uniform 300;

boundaryField
{
  maxX
  {
    type zeroGradient;
  }

  minZ
  {
    type zeroGradient;
  }

  maxZ
  {
    type zeroGradient;
  }
```



チュートリアルの
system/rightSolid/changeDictionayDictを参照にして設定する

```
rightSolid_to_bottomAir
{
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
}

rightSolid_to_topAir
{
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
}

rightSolid_to_heater
{
        type            compressible::turbulentTemperatureCoupledBaffleMixed;
        value           uniform 300;
        neighbourFieldName T;
        kappa           solidThermo;
        kappaName       none;
}
}       // ************************************************************************
```

● 0/rightSolid/pの修正

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  maxX
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  rightSolid_to_bottomAir
  {
    type mappedWall;
  }
  rightSolid_to_topAir
  {
    type mappedWall;
  }
  rightSolid_to_heater
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 1 -1 -2 0 0 0 0 ];

internalField uniform 100000;

boundaryField
{
  maxX
  {
        type            calculated;
        value           uniform 100000;
  }

  minZ
  {
        type            calculated;
        value           uniform 100000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 100000;
  }
```

```
  rightSolid_to_bottomAir
  {
        type            calculated;
        value           uniform 0;
  }

  rightSolid_to_topAir
  {
        type            calculated;
        value           uniform 0;
  }

  rightSolid_to_heater
  {
        type            calculated;
        value           uniform 0;
  }
}       // ************************
```

● 0/rightSolid/rhoの修正

```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  maxX
  {
    type zeroGradient;
  }
  minZ
  {
    type zeroGradient;
  }
  maxZ
  {
    type zeroGradient;
  }
  rightSolid_to_bottomAir
  {
    type mappedWall;
  }
  rightSolid_to_topAir
  {
    type mappedWall;
  }
  rightSolid_to_heater
  {
    type mappedWall;
  }
}       // ********************
```

```
dimensions [ 1 -3 0 0 0 0 0 ];

internalField uniform 8000;

boundaryField
{
  maxX
  {
        type            calculated;
        value           uniform 8000;
  }

  minZ
  {
        type            calculated;
        value           uniform 8000;
  }

  maxZ
  {
        type            calculated;
        value           uniform 8000;
  }
```

```
  rightSolid_to_bottomAir
  {
        type            calculated;
        value           uniform 0;
  }

  rightSolid_to_topAir
  {
        type            calculated;
        value           uniform 0;
  }

  rightSolid_to_heater
  {
        type            calculated;
        value           uniform 0;
  }
}       // ************************
```

● 修正ファイルの保存

修正したファイル群は，マクロで計算を実行させるため保存する

● constant/bottomAir/polyMesh/boundaryの修正

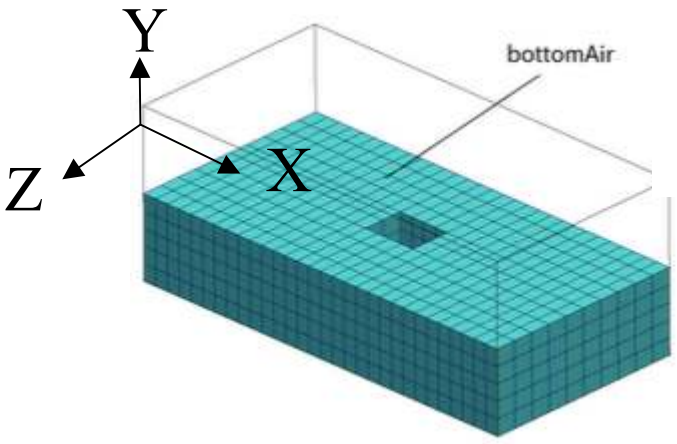
changeDictionaryDictを参照して変更する

```
boundary
{
    minX
    {
        type            wall;
    }
    maxX
    {
        type            wall;
    }
}
```

```
8
(
    minX
    {
        type            patch;
        nFaces          128;
        startFace       17022;
    }
    maxX
    {
        type            patch;
        nFaces          119;
        startFace       17150;
    }
    minY
    {
        type            wall;
        nFaces          1168;
        startFace       17269;
    }
```

```
8
(
    minX
    {
        type            wall;
        nFaces          128;
        startFace       17022;
    }
    maxX
    {
        type            wall;
        nFaces          119;
        startFace       17150;
    }
    minY
    {
        type            wall;
        nFaces          1168;
        startFace       17269;
    }
    minZ
```

２箇所を変更する



● constant/leftSolid/polyMesh/boundaryの修正

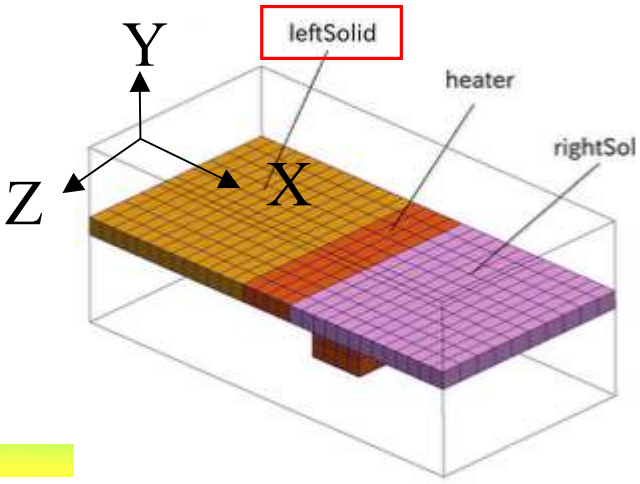
changeDictionaryDictを参照して変更する

```
boundary
{
    minZ
    {
        type            patch;
    }
    maxZ
    {
        type            patch;
    }
}
```

```
6
(
minX
{
    type            patch;
    nFaces          40;
    startFace       2508;
}
minZ
{
    type            wall;
    nFaces          52;
    startFace       2548;
}
maxZ
{
    type            wall;
    nFaces          52;
    startFace       2600;
}
```

```
6
(
minX
{
    type            patch;
    nFaces          40;
    startFace       2508;
}
minZ
{
    type            patch;
    nFaces          52;
    startFace       2548;
}
maxZ
{
    type            patch;
    nFaces          52;
    startFace       2600;
}
```

２箇所を変更する

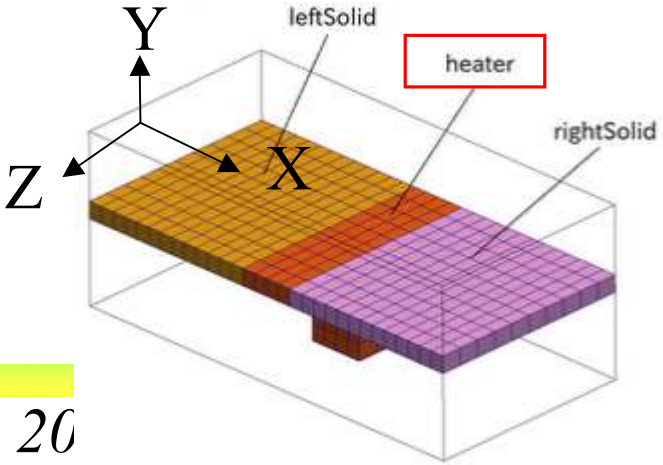


● constant/heater/polyMesh/boundaryの修正

changeDictionaryDictを参照して変更する

```
boundary
{
    minY
    {
        type            patch;
    }
    minZ
    {
        type            patch;
    }
    maxZ
    {
        type            patch;
    }
}
```

3箇所を変更する




```
7
(
minY
{
    type            wall;
    nFaces          32;
    startFace       1584;
}
minZ
{
    type            wall;
    nFaces          16;
    startFace       1616;
}
maxZ
{
    type            wall;
    nFaces          16;
    startFace       1632;
}
```

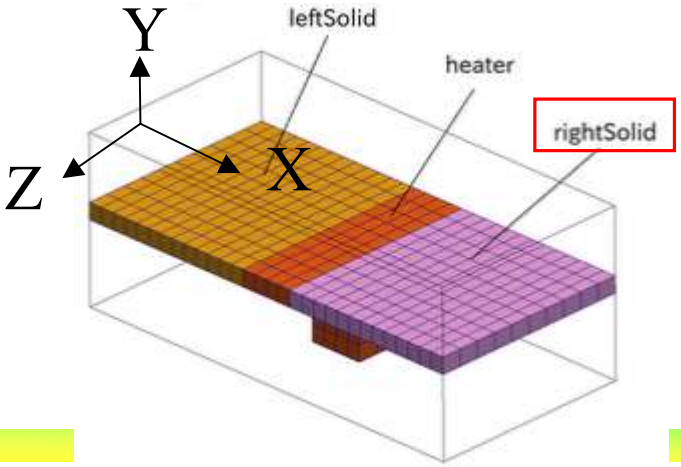
```
7
(
minY
{
    type            patch;
    nFaces          32;
    startFace       1584;
}
minZ
{
    type            patch;
    nFaces          16;
    startFace       1616;
}
maxZ
{
    type            patch;
    nFaces          16;
    startFace       1632;
}
```

● constant/rightSolid/polyMesh/boundaryの修正

changeDictionaryDictを参照して変更する

```
boundary
{
    minZ
    {
        type            patch;
    }
    maxZ
    {
        type            patch;
    }
}
```

２箇所を変更する

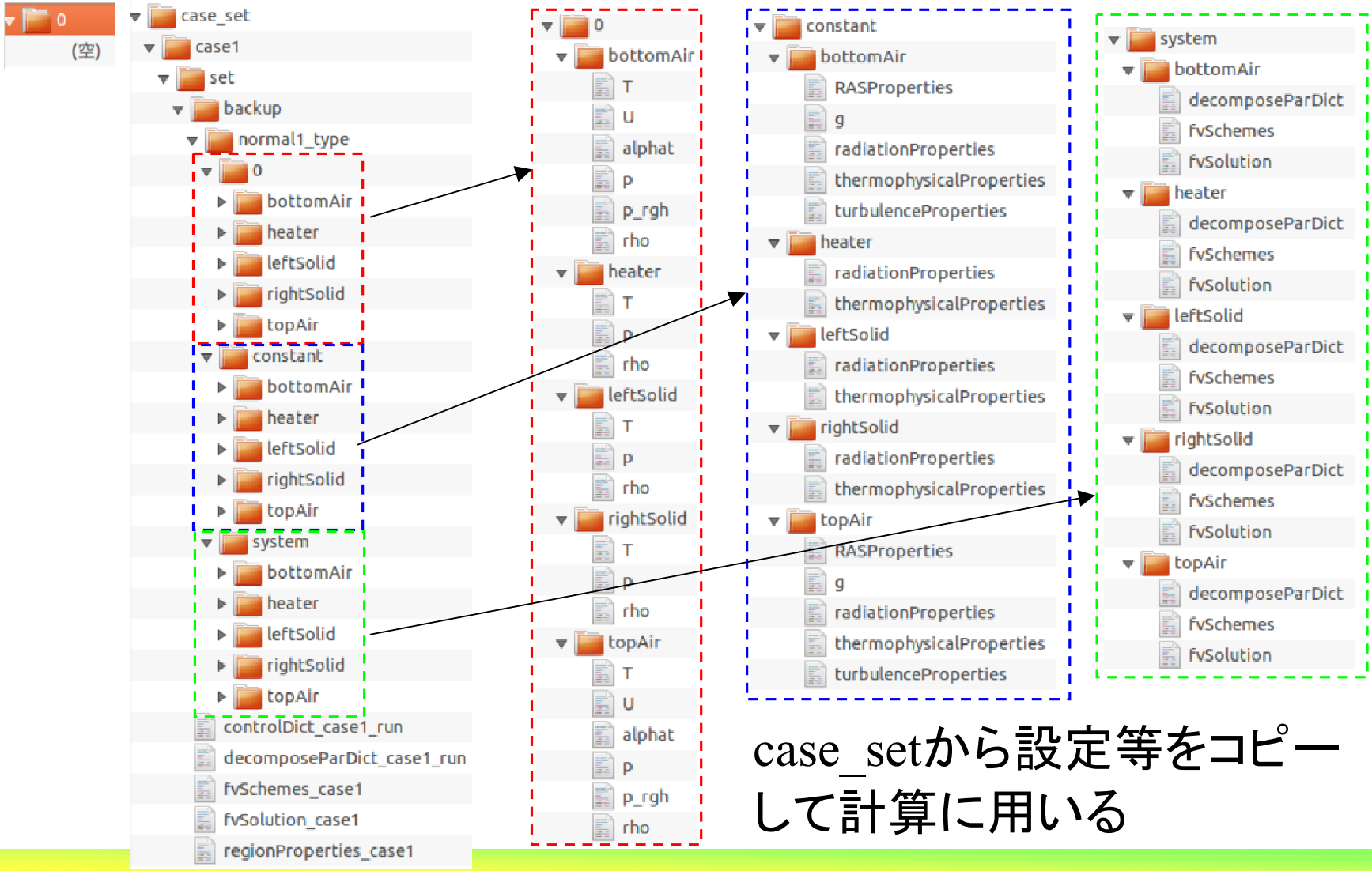



```
6
(
maxX
{
    type            patch;
    nFaces          40;
    startFace       2508;
}
minZ
{
    type            wall;
    nFaces          52;
    startFace       2548;
}
maxZ
{
    type            wall;
    nFaces          52;
    startFace       2600;
}
```

```
6
(
maxX
{
    type            patch;
    nFaces          40;
    startFace       2508;
}
minZ
{
    type            patch;
    nFaces          52;
    startFace       2548;
}
maxZ
{
    type            patch;
    nFaces          52;
    startFace       2600;
}
```

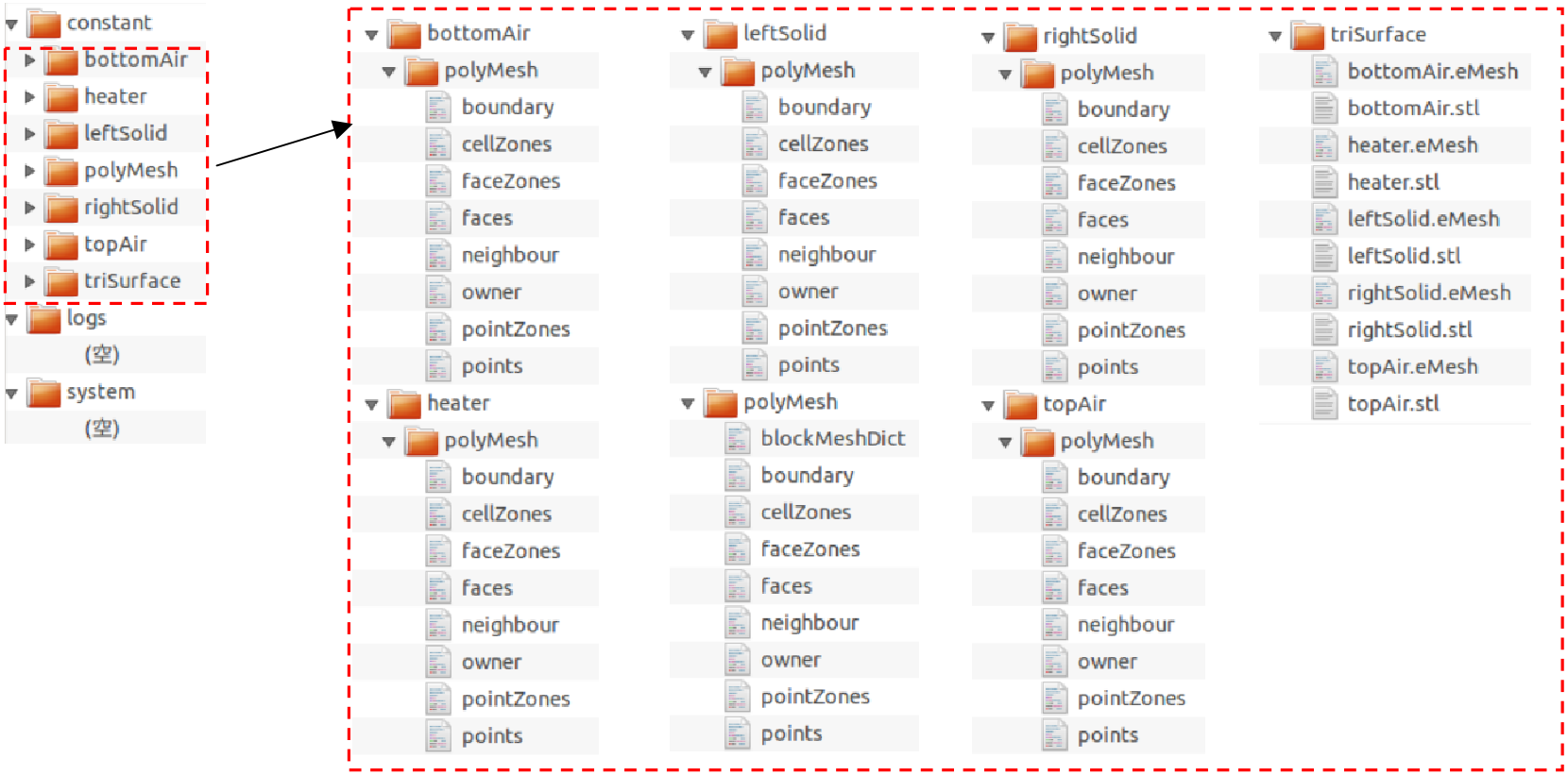
3. コマンドによる計算実施

● コマンドによる計算実施前のホルダ整理



case_setから設定等をコピーして計算に用いる

● コマンドによる計算実施前のホルダ整理



こんな感じで整理をしました

● コマンドによる計算実施手順

①ファイルの設定　変数名を設定する

```
CASE_DIR=case_set/case1/set
UP_DIR_0=case_set/case1/set/backup/normal1_type/0
UP_DIR_constant=case_set/case1/set/backup/normal1_type/constant
UP_DIR_system=case_set/case1/set/backup/normal1_type/system

DECOMPOSE_PAR=decomposeParDict_case1_run
CONTROL_DICT_run=controlDict_case1_run
REGION=regionProperties_case1
FV_SCH=fvSchemes_case1
FV_SOL=fvSolution_case1

REGION_FLOW_1=bottomAir
REGION_FLOW_2=topAir
REGION_SOLID_1=heater
REGION_SOLID_2=leftSolid
REGION_SOLID_3=rightSolid
```

②計算に必要なホルダ，ファイルの設定

```
rm -f -r 0
mkdir 0
cp -r $UP_DIR_0/* 0/
cp -r $UP_DIR_constant/* constant/
cp -r $UP_DIR_system/* system/
cp -r $CASE_DIR/$REGION constant/regionProperties
cp -r $CASE_DIR/$FV_SCH system/fvSchemes
cp -r $CASE_DIR/$FV_SOL system/fvSolution
cp -r $CASE_DIR/$CONTROL_DICT_run system/controlDict
cp -r $CASE_DIR/$DECOMPOSE_PAR system/decomposeParDict
```

③並列計算のための分割（ここでは4並列）

```
$runApplication decomposePar -allRegions > logs/log.decomposePar
```

④分割した領域をリナンバリングする

```
for ii in processor0 processor1 processor2 processor3
do
   for jj in $REGION_FLOW_1 $REGION_FLOW_2 $REGION_SOLID_1 $REGION_SOLID_2 $REGION_SOLID_3
   do
      rm -f -r $ii/constant/$jj/system
      mkdir    $ii/constant/$jj/system
      rm -f -r $ii/temp
      mkdir $ii/temp
      mkdir $ii/temp/0
      mkdir $ii/temp/constant
      mkdir $ii/temp/constant/polyMesh
      mkdir $ii/temp/system

      cp $CASE_DIR/$CONTROL_DICT_run $ii/constant/$jj/system/controlDict
      cp -r $ii/0/$jj/*                  $ii/temp/0
      cp -r $ii/constant/$jj/polyMesh/* $ii/temp/constant/polyMesh
      cp -r $CASE_DIR/$CONTROL_DICT_run $ii/temp/system/controlDict
      cp -r $CASE_DIR/$FV_SCH           $ii/temp/system/fvSchemes
      cp -r $CASE_DIR/$FV_SOL           $ii/temp/system/fvSolution

      renumberMesh -case $ii/temp -overwrite

      cp -r $ii/temp/constant/polyMesh/* $ii/constant/$jj/polyMesh
      rm -f -r $ii/temp
   done
done
```

作業用のホルダ設定

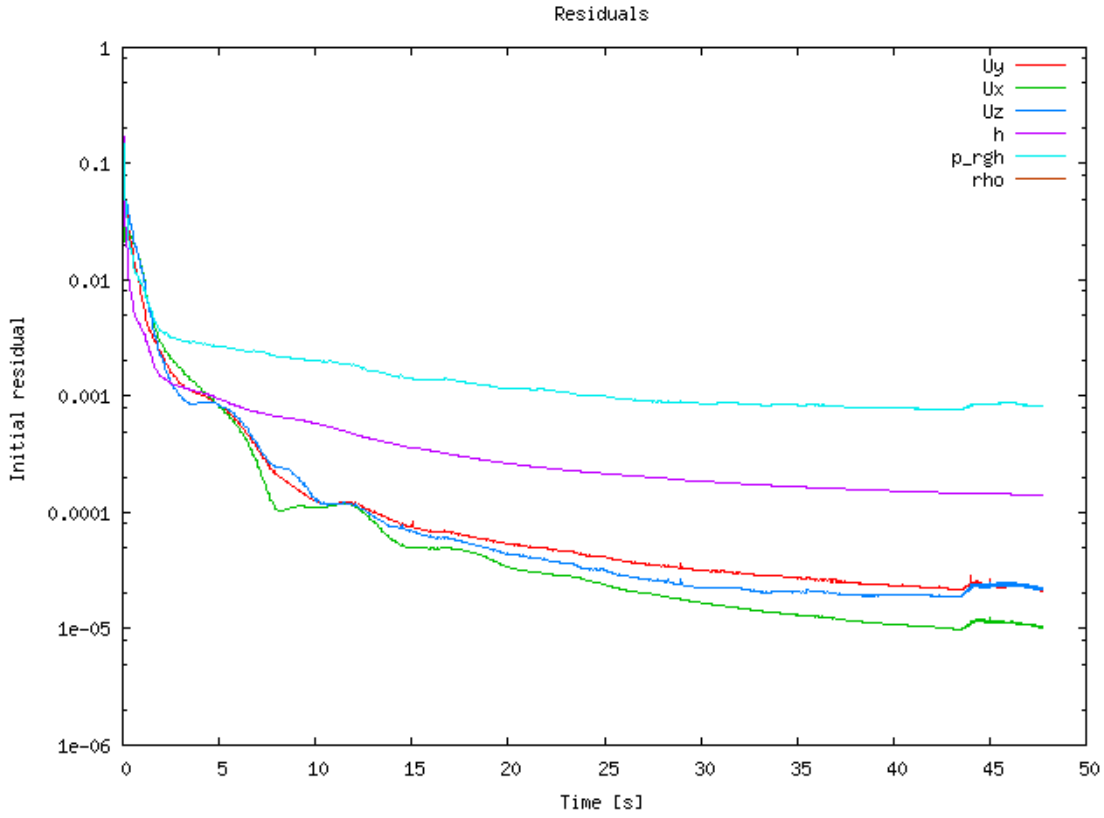
必要なファイルのコピー

リナンバリング

処理ファイルを戻す

⑤4並列での計算

```
pyFoamPlotRunner.py mpirun -np 4 chtMultiRegionFoam -parallel > logs/log.chtMultiRegionFoam_laminar
```



⑥ホルダの再構築

```
$runApplication reconstructPar -allRegions > logs/log.reconstructPar
```

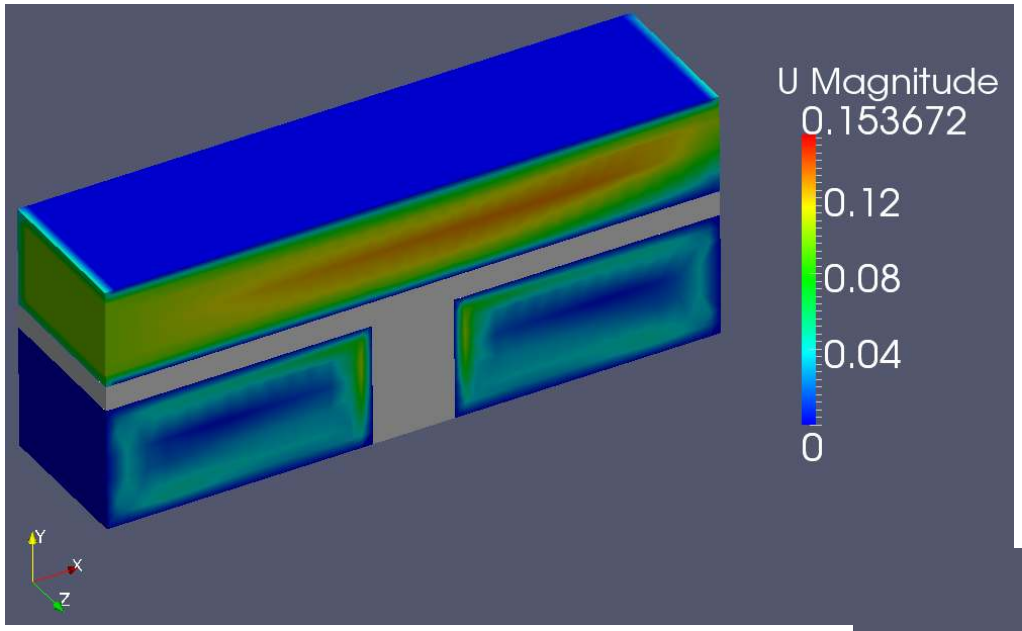
⑦並列計算で利用したホルダの削除

```
rm -r processor0
rm -r processor1
rm -r processor2
rm -r processor3
```

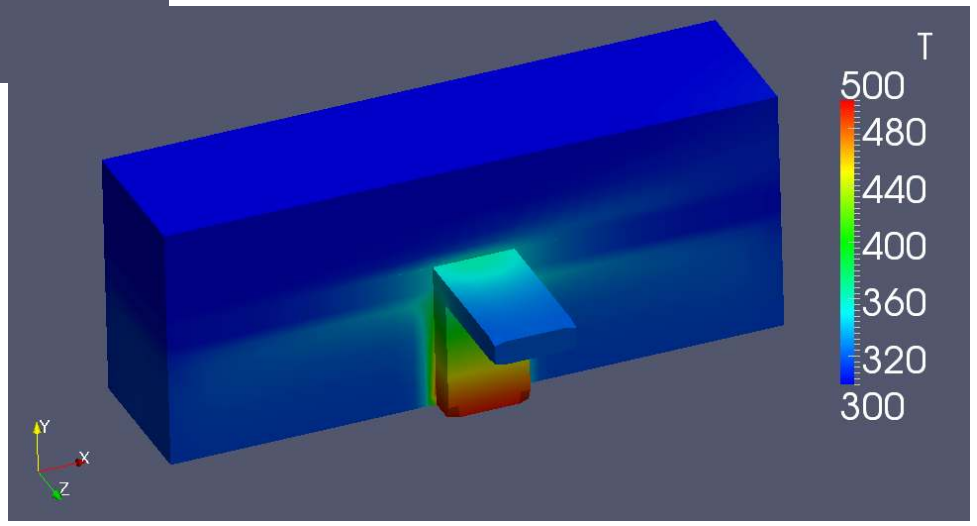
⑧計算結果の表示準備

```
paraFoam -touchAll
```

● 計算結果



チュートリアルの
結果と同じになる



４．まとめ，その他

●まとめ，その他

- snappyMultiRegionHeaterチュートリアルの中身を観察し，メッシュ，計算の過程をチェックした。
- コマンド，手修正によりメッシュの作成，計算の実施を行った。
- simpleFoamの計算に比べて，設定事項，修正事項が多い。
- 今後の予定としては，RAS，LESで計算実施。