新TreeFoamの紹介

chtMultiRegionFoamをにらんで
TreeFoamに機能追加
→　regionが増えても処理ができるように

1. csvファイルによるメッシュ作成
2. csvファイルによるsetFields
3. region分割、計算開始
4. まとめ

14/02/15
藤井　成樹

TreeFoam外観の変更

TreeFoam上で
startFrom、stopAtが設定できる

TreeFoam_2.11-140119 (0)

ファイル(F) case作成変更(M) 編集(E) 計算(C) ツール(T) ヘルプ(H)

case directory: ...aeuser/CAE/CAE-FOAM/fluidSolid-test/fluidSolid-mesh_copy0 OpenFoam環境: bashrc-FOAM-2.2.2

現在の解析case名: regCase-cht

solver: chtMultiRegionFoam

startFrom: latestTime | stopAt: endTime:2 | controlDict: 編集

formatや並列数を表示
anP ：ascii、非圧縮、シングル処理
BCP2 ：binary、圧縮、2並列

Tree		BCPn	nR	st	ed
CAE-FOAM					
fluidManySolid					
fluidSolid-test					
fluidSolid-mesh	icoFoam	anP	1	0.0	
fluidSolid-mesh_copy0	chtMultiRegionFoam	BnP	1	0.0	
regCase-cht	chtMultiRegionFoam	BnP2	2	0.0	0.1
regCase-chtQv	chtQvMultiRegionFoam	BnP2	1	0.0	

log open /home/caeuser/TreeFoam/temp/0_logTreeFoam

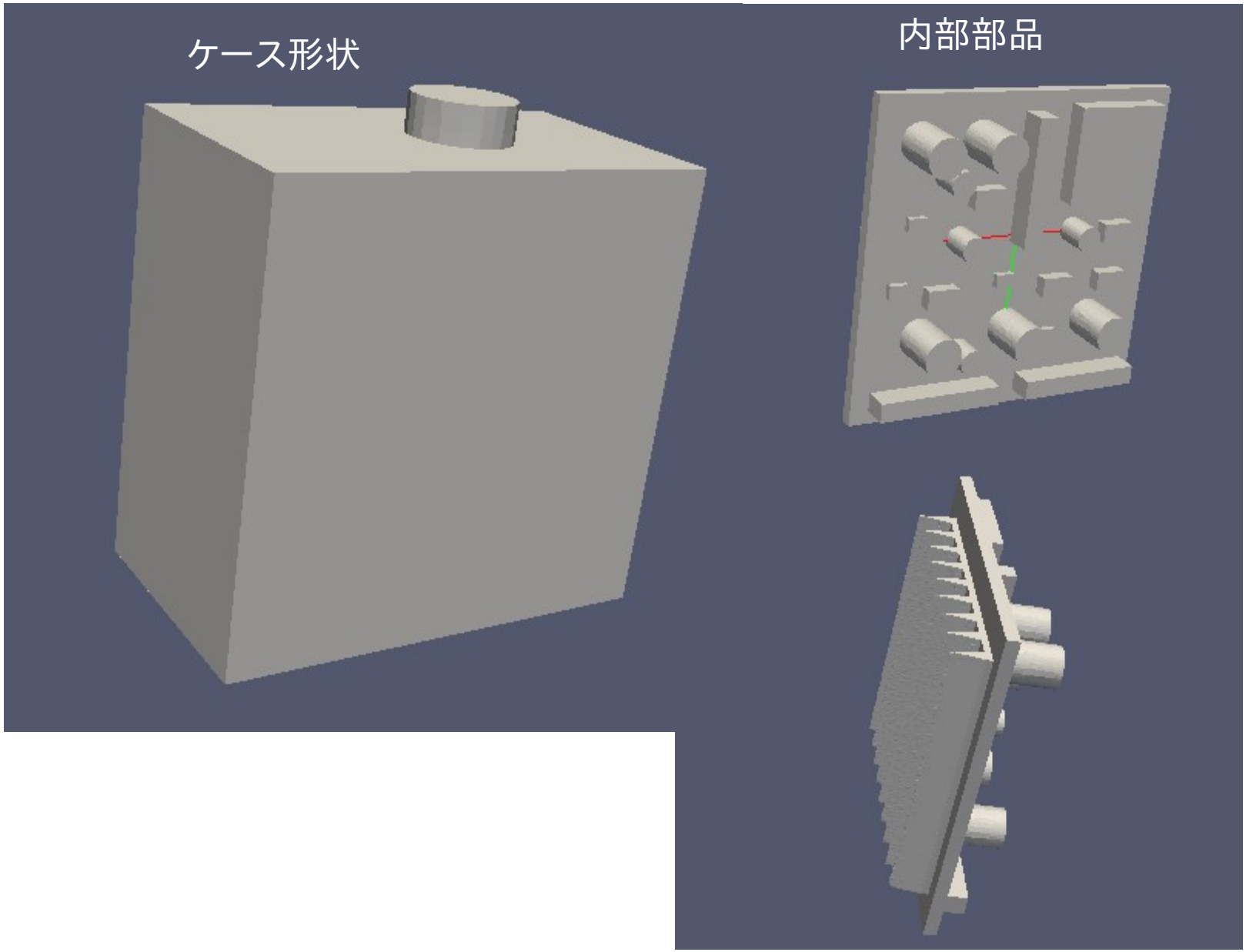
TreeFoam ver 2.11-140119 (0) を起動しました。
OpenFOAM - 2.2.2

windowを分割してlogを表示
logを表示するかしないかは、configTreeFoamで設定

TreeFoamのアイコンも準備

多くのregionを含むモデルの解析が容易にできる様に･･･

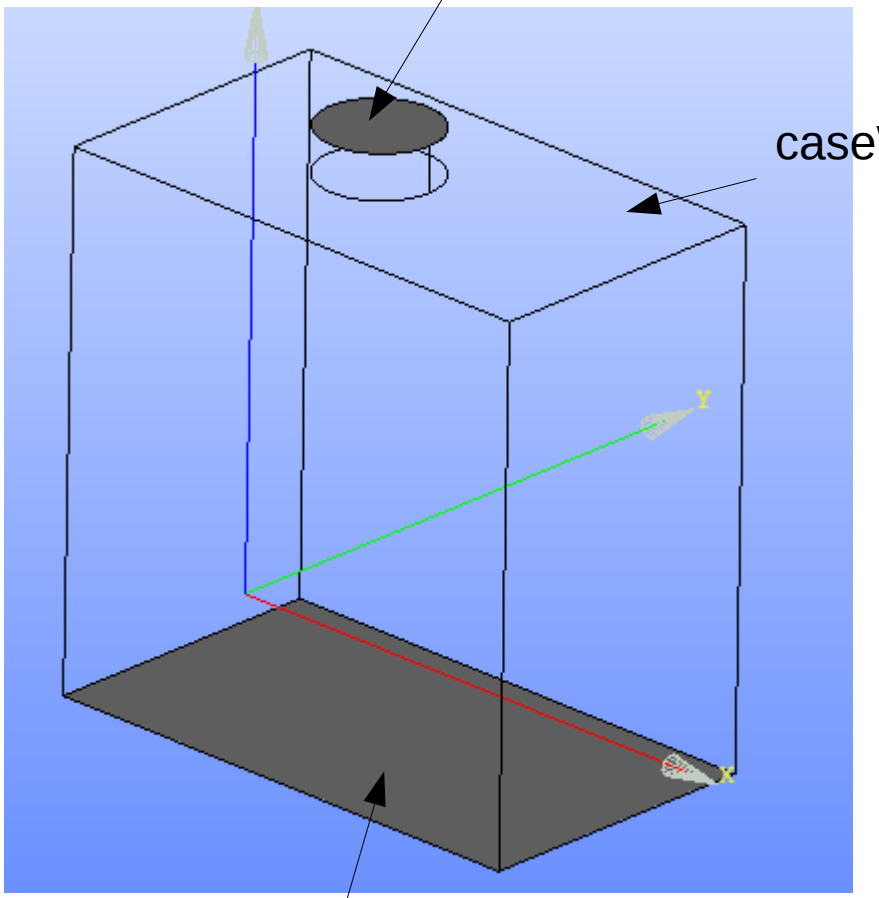
部品点数：25ヶ



ベースに多数の
部品配置

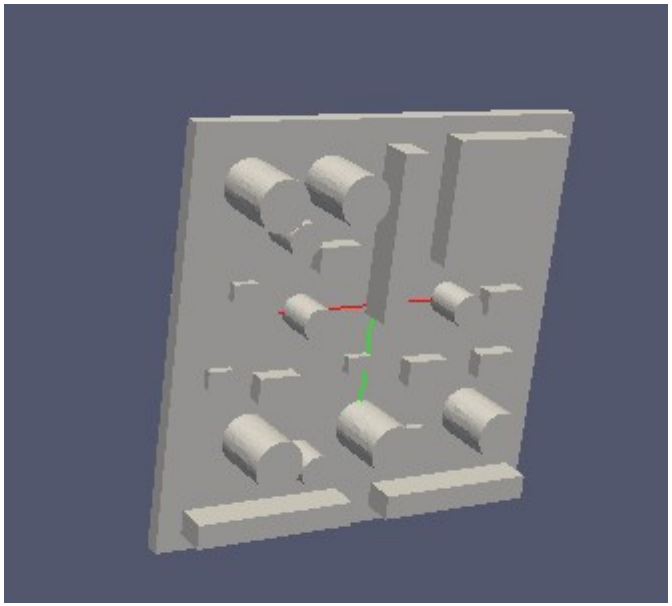
ベース裏面に
ヒートシンク配置

計算条件の設定内容



case上面の排気面（outW）から1m/sで排気し、case下面全体（inW）から吸気する構造。

case内部には、発熱部品を配置



計算条件（内部部品）

各部品の材質と発熱量

No	領域名	材質	発熱量（W）	備考
1	air	空気	0	発熱なし
2	base	Al	20	
3	cyli1	Al	10	
4	cyli2	Fe	2	
5	cyli3	Al	5	
6	cyli4	Al	5	
7	cyli5	mold	0	発熱なし
8	cyli6	Al	1	
9	cyli7	Al	2	
10	cyli8	Fe	1	
11	cyli9	Fe	1	
12	heatSink	Al	0	発熱なし
13	rect1	mold	0	発熱なし
14	rect10	Al	1	

No	領域名	材質	発熱量（W）	備考
15	rect11	Cu	1	
16	rect12	Al	5	
17	rect13	mold	2	
18	rect2	ceramic	0	発熱なし
19	rect3	Al	1	
20	rect4	Al	3	
21	rect5	Cu	1	
22	rect6	mold	1	
23	rect7	mold	1	
24	rect8	Al	20	
25	rect9	Al	10	

部品（固体）の発熱が計算できる様に
chtMultiRegionFoamを修正

1.　メッシュ作成

snappyHexMeshDictを直接編集せずに、csvファイルを準備して、メッシュを作成。

新しく作成したcase
tutorials/heatTransfer/chtMultiRegionFoam/multiRegionHeaterをコピー
（Allrunを実行してcaseを完成しておく）

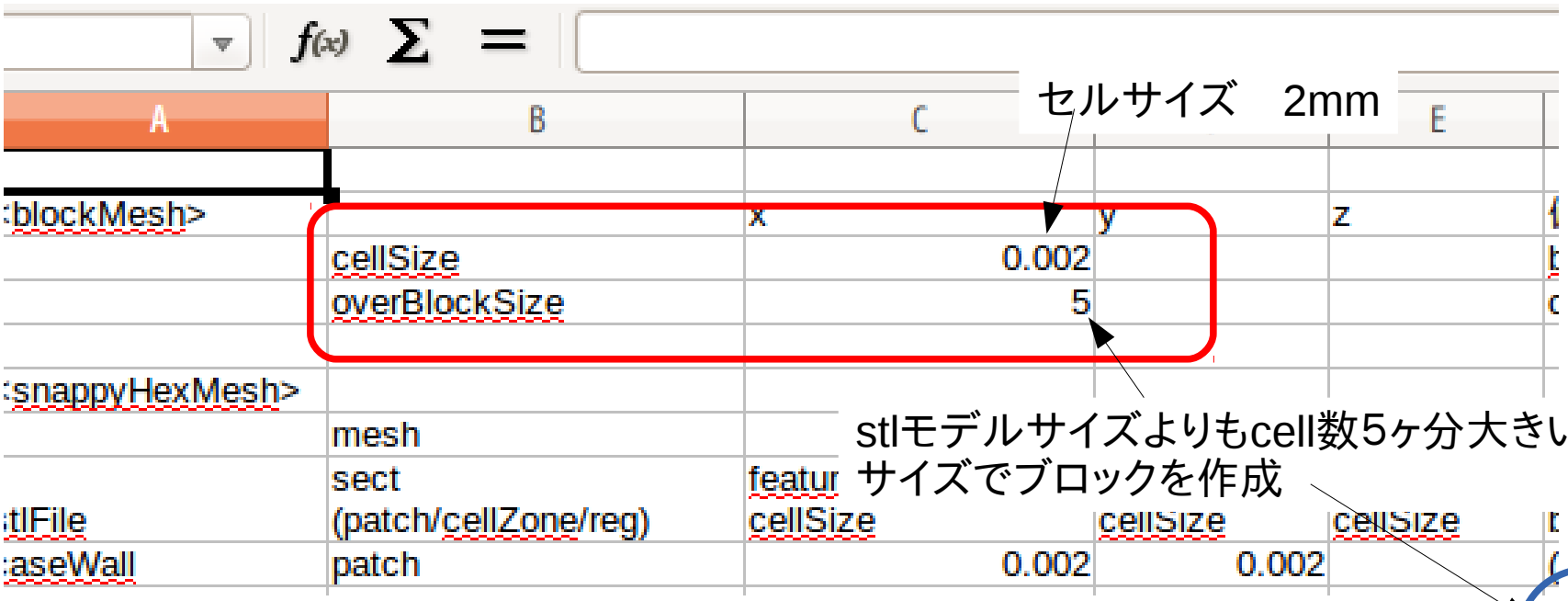
デフォルトのcsvファイル

blockMeshの設定

snappyHexMeshの設定

	A			D	E	F
1						
2	<blockMesh>		x	y	z	備考
3		cellSize	0.0023	0.0013	0.0025	blockMeshのcellSize
4		overBlockSize	5	5	5	cells: stlのMinMax値を越えるcell数
5						
6	<snappyHexMesh>					
7		mesh	0.025	0	0.028	meshの位置(materialPoint)
8	stlFile	sect (patch/cellZone/reg)	featureEdge cellSize	base cellSize	fine cellSize	featureEdge: cellSizeを入力したstlのみ抽出。base: surface, regionとも設定する。
9	base					(0.05 0.002 0.05)
10	caseWall					(0.07 0.04 0.076)
11	cyli1					(0.006 0.008 0.006)
12	cyli2					(0.006 0.008 0.006)
13	cyli3					(0.006 0.008 0.006)
14	cyli4					(0.004 0.005 0.004)
15	cyli5					(0.004 0.005 0.004)
16	cyli6					(0.004 0.005 0.004)
17	cyli7					(0.006 0.008 0.006)
18	cyli8					(0.006 0.008 0.006)
19	cyli9					(0.004 0.005 0.004)
20	heatSink					(0.042 0.006 0.04)
21	inW					(0.07 0.04 0.0)
22	outW					(0.016 0.016 0.0)
23	rect1					(0.02 0.004 0.003)
24	rect10					(0.003 0.001 0.002)
25	rect11					(0.015 0.003 0.015)
26	rect12					(0.005 0.002 0.003)
27	rect13					(0.003 0.001 0.002)
28	rect2					(0.02 0.004 0.003)
29	rect3					(0.005 0.002 0.003)

blockMeshの設定



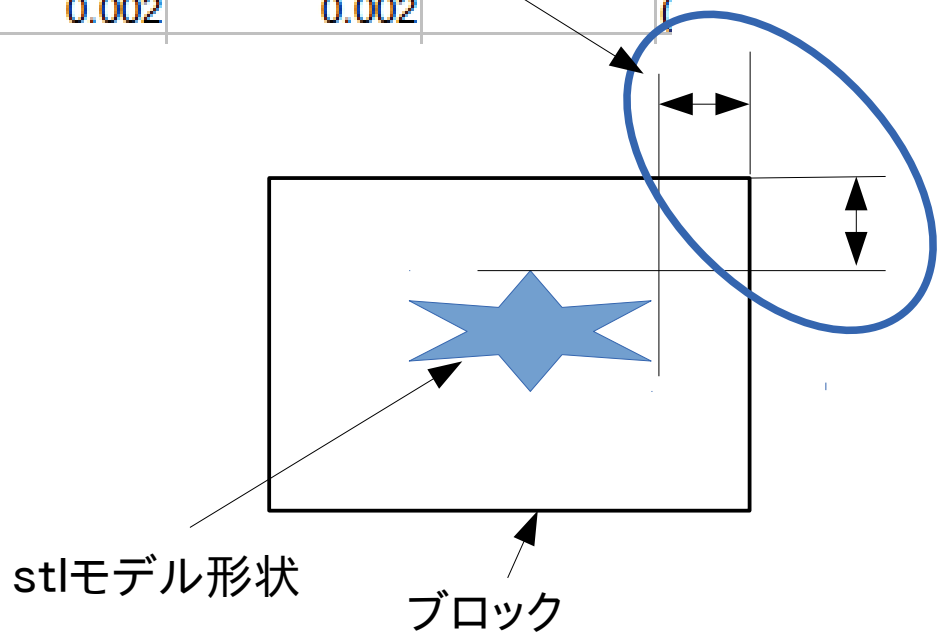
A	B	C	D	E
:blockMesh>		x	y	z
	cellSize	0.002		
	overBlockSize	5		
:snappyHexMesh>				
	mesh			
	sect	featur		
:tlFile	(patch/cellZone/reg)	cellSize	cellSize	cellSize
:aseWall	patch	0.002	0.002	

セルサイズ　2mm

stlモデルサイズよりもcell数5ヶ分大きい
サイズでブロックを作成

通常：blockの座標とXYZの分割数を入力
→　座標を意識する必要がある
セルサイズが直感的にわかりにくい

座標を意識すること無く、セルサイズを直接入力



snappyHexMeshの設定

mesh内のポイント
（デフォルトで中心座標が設定）

<snappyHexMesh>					
	mesh	0.025	0	0.025	meshの位置(materia
stlFile	sect (patch/cellZone/reg)	featureEdge cellSize	base cellSize	fine cellSize	featureEdge: cellSiz base: surface, regior
caseWall	patch	0.002	0.002		(0.07 0.04 0.076)
inW	patch	0.002	0.002		(0.07 0.04 0.0)
outW	patch	0.002	0.002	0.001	(0.016 0.016 0.0)
base	cellZone	0.001	0.001		(0.05 0.002 0.05)
cyli1	cellZone	0.001	0.001	0.0005	(0.006 0.008 0.006)
cyli2	cellZone	0.001	0.001	0.0005	(0.006 0.008 0.006)
cyli3	cellZone	0.001	0.001	0.0005	(0.006 0.008 0.006)
cyli4	cellZone	0.001	0.001	0.0005	(0.004 0.005 0.004)
cyli5	cellZone	0.001	0.001	0.0005	(0.004 0.005 0.004)
cyli6	cellZone	0.001	0.001	0.0005	(0.004 0.005 0.004)

stlファイル名

区分
- patch
- cellZone
- reg（領域を定義：部分的にcellサイズを変更する場合に使用）

特徴線の抽出
- 空白：抽出しない
- 値：抽出する

基本のcellサイズ
表面、内部共

最小cellサイズ
表面のcell

直接cellサイズを入力するように設定
（blockMeshサイズが変わっても影響を受けない）
データ入力後、csv形式で保存する。

blockMeshDict、snappyHexMeshDictの作成、mesh作成

ファイル名を選択している事を確認後、

クリックして、
Dictファイル、mesh作成

特徴線を抽出るために、
「eMesh」ファイルが必要だが、
無くても、自動で作成してくれるので
意識して作成する必要はない。

ここまでの操作で、blockMeshDict、snappyHexMeshDictができあがり
snappyHexMeshを実行してメッシュを作成する

できあがったblockMeshDict、snappyHexMeshDict

BlockMeshDict
元々のfileを修正

blockMeshDict

```
5 |   \\  /    A nd           | Web:      www.OpenFOAM.org
6 |    \\/     M anipulation  |
7 \*---------------------------------------------------------------------------
8 FoamFile
9 {
10     version     2.0;
11     format      ascii;
12     class       dictionary;
13     object      blockMeshDict;
14 }
15 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
16
17 convertToMeters 1;
18
19 vertices
20     (
21         ( -0.02 -0.03 -0.02 )
22         ( 0.07 -0.03 -0.02 )
23         ( 0.07 0.03 -0.02 )
24         ( -0.02 0.03 -0.02 )
25         ( -0.02 -0.03 0.076 )
26         ( 0.07 -0.03 0.076 )
27         ( 0.07 0.03 0.076 )
28         ( -0.02 0.03 0.076 )
29     );
30
31 blocks
32     (
33     hex (0 1 2 3 4 5 6 7) (45 30 48) simpleGrading (1 1 1)
34     );
35
36 edges
37 (
```

座標、分割数を
修正

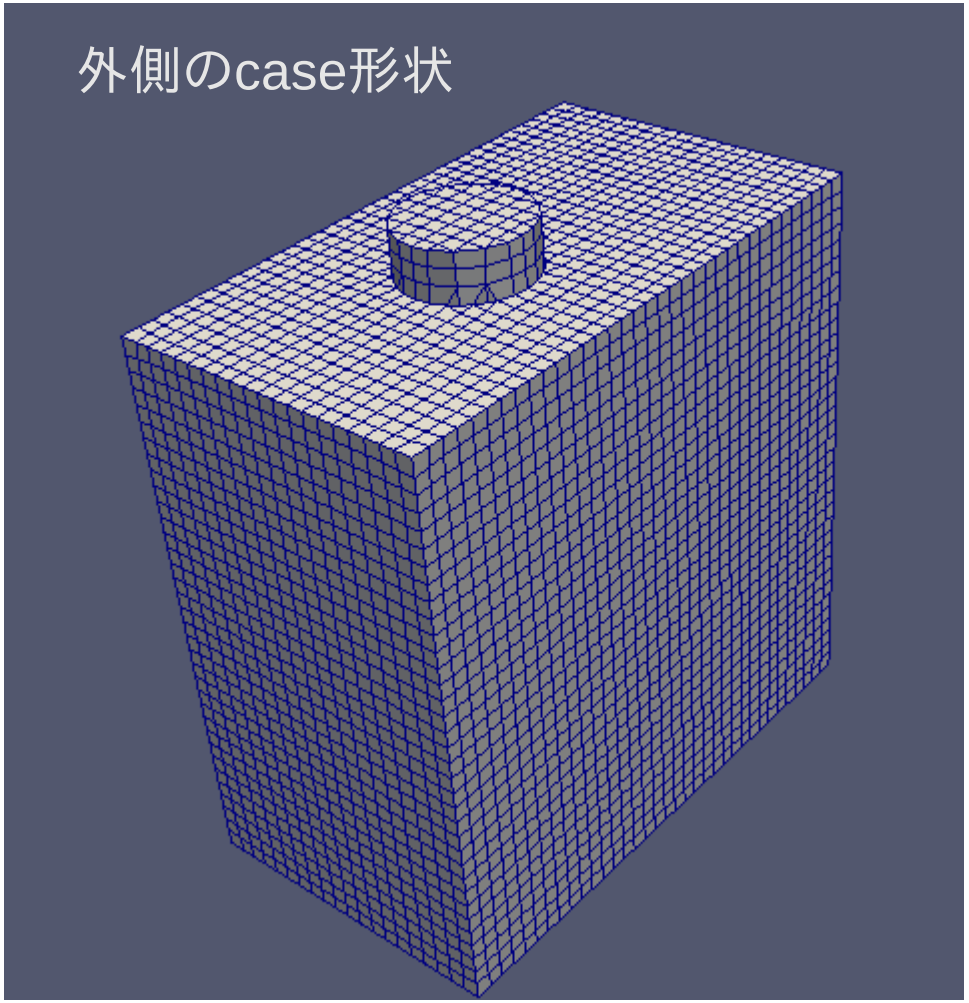
snappyHexMeshDict
新たに作成

blockMeshDict | snappyHexMeshDict

```
1 /*--------------------------------*- C++ -*----------------------------------
2 |       o          |
3 |    o     o       | HelyxOS
4 |   o   O   o      | Version: 1.0.1
5 |    o     o       | Web:      http://www.engys.com
6 |       o          |
7 \*---------------------------------------------------------------------------
8 FoamFile
9 {
10     version 2.0;
11     format ascii;
12     class dictionary;
13     location system;
14     object snappyHexMeshDict;
15 }
16
17     castellatedMesh true;
18     snap true;
19     addLayers true;
20     geometry
21     {
22         caseWall.stl
23         {
24             type triSurfaceMesh;
25             name caseWall;
26         }
27
28         inW.stl
29         {
30             type triSurfaceMesh;
31             name inW;
32         }
33
```

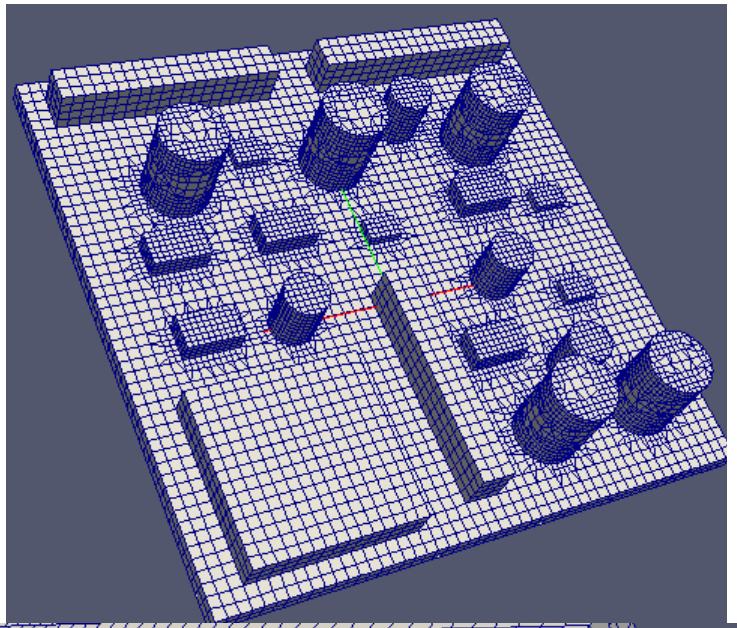
stlファイルが追加
されている。

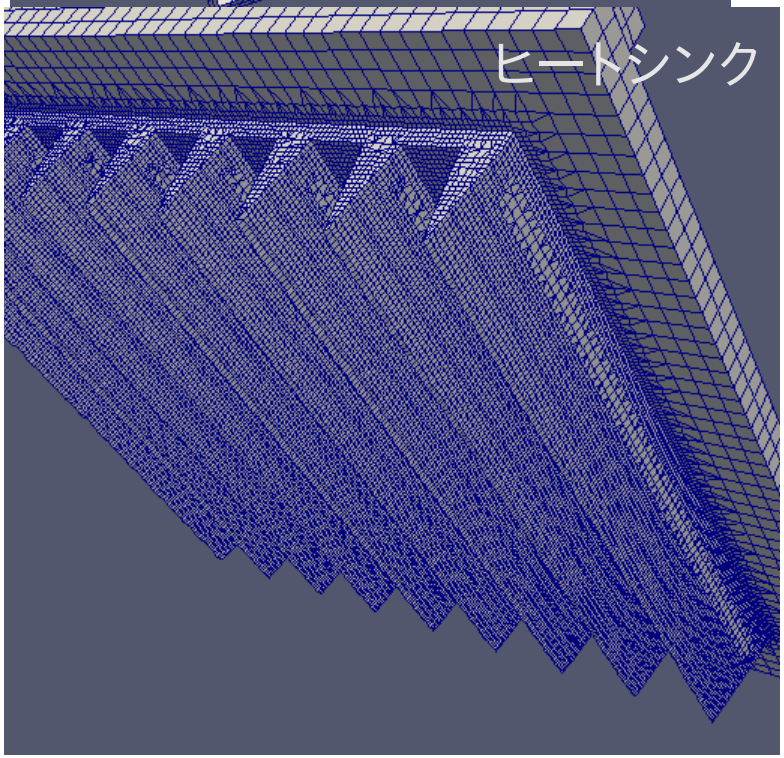
出来上がったmesh



csvファイルを基本にして
できあがったmesh

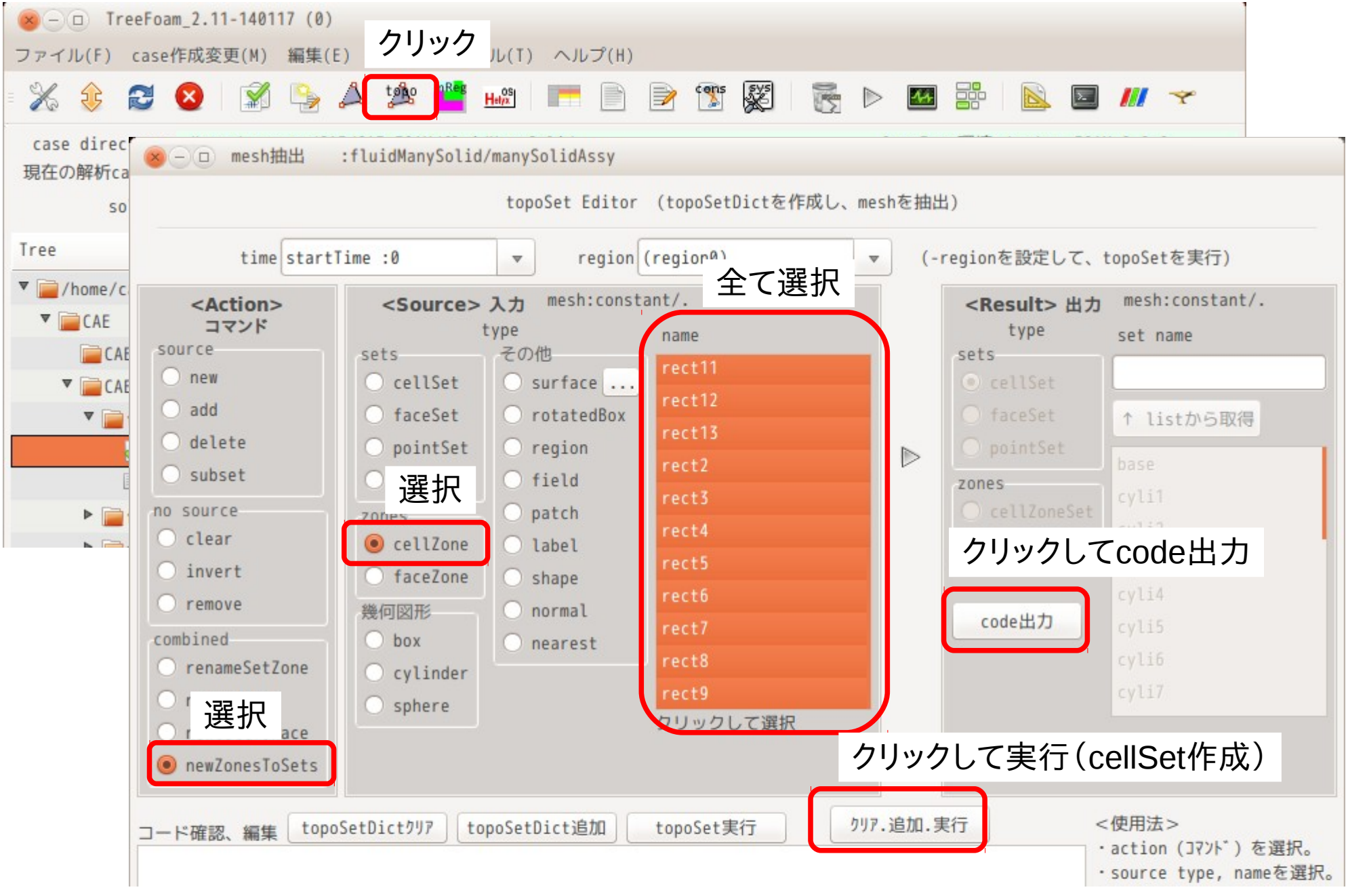
内部部品





2.　setFields（csvファイルによる）

mesh作成時、できあがった全てのcellZoneからcellCetを作成する。



setFieldsDictを作成

TreeFoam_2.11-140117 (0)

ファイル(F) case作成変更(M) 編集(E) 計算(C) ツール(T) ヘルプ

クリック

OpenFoam環境: bashrc-FOAM-2.2.2

startFrom stopAt controlDict

startTime:0 endTime:1 編集

BCPn nR st ed

pleFoam BnP 1 0.0

Fieldへのデータセット

timeFolder内の各Fieldへのデータセット（クリア）

編集するfolder

time region

startTime :0 (region0)

全てのcellSetを選択

dataセットするFieldを選択

cellSets

rect5

rect6

rect7

rect8

rect9

fields

Qw

T

U

alphat

epsilon

cellSet作成 topoSetEditor起動 field内容確認 gridEditor起動

setFieldsDict作成...

クリックするとsetFields作成画面が現れる。

setFields実行

setFieldsDict編集 setFields実行... systemフォルダ開く

setFieldsを実行して
dataをfieldにセット

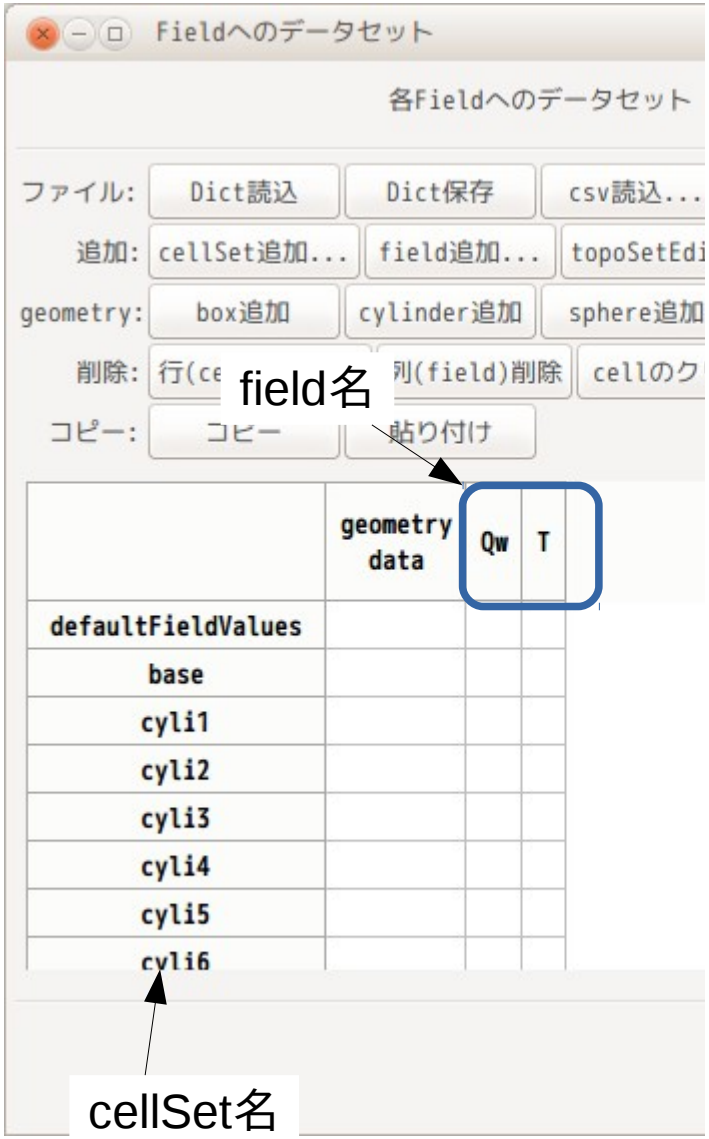
fieldのクリア

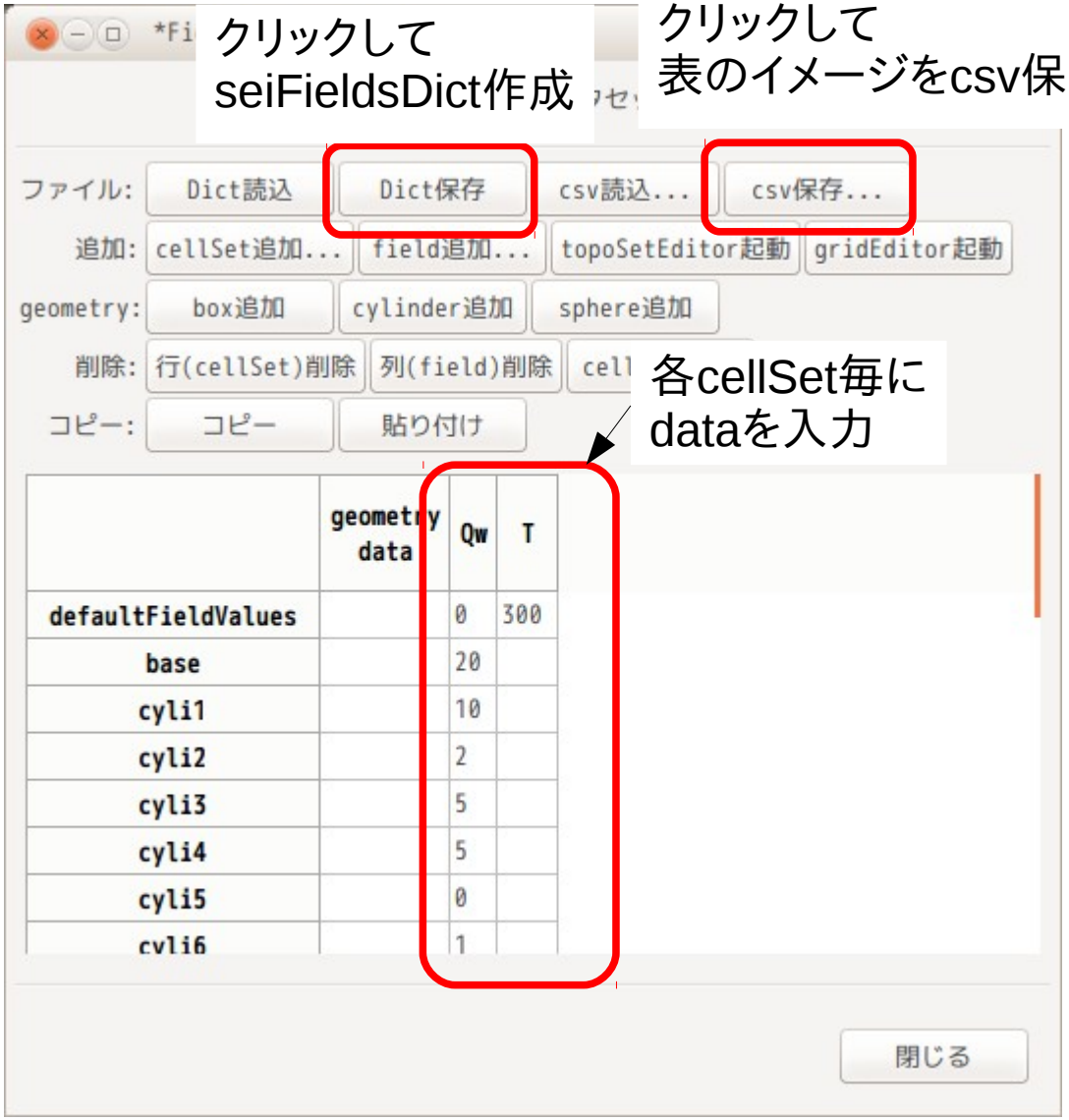
internalField...

閉じる

setFields画面上で
setFieldsDictファイルを作成する
作成後、この画面上で、setFields
を実行する。

setFieldsDict作成画面





Qwフィールドの各領域（cellSet）にワット数を設定
Tフィールドの初期値を設定

できあがったsetFieldsDict

d/manySolidAssy/system)

ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

開く 保存 元に戻す

setFieldsDict

```
17
18
19 defaultFieldValues
20 (
21     volScalarFieldValue Qw 0
22     volScalarFieldValue T 300
23 );
24
25 regions
26 (
27     cellToCell
28     {
29         set base;
30         fieldValues
31         (
32             volScalarFieldValue Qw 20
33         );
34     }
35
36     cellToCell
37     {
38         set cyli1;
39         fieldValues
40         (
41             volScalarFieldValue Qw 10
42         );
43     }
44
45     cellToCell
46     {
47         set cyli2;
48         fieldValues
49         (
```

C ▾ タブの幅:: 4

csv保存した内容
（表のイメージをそのまま保存する）

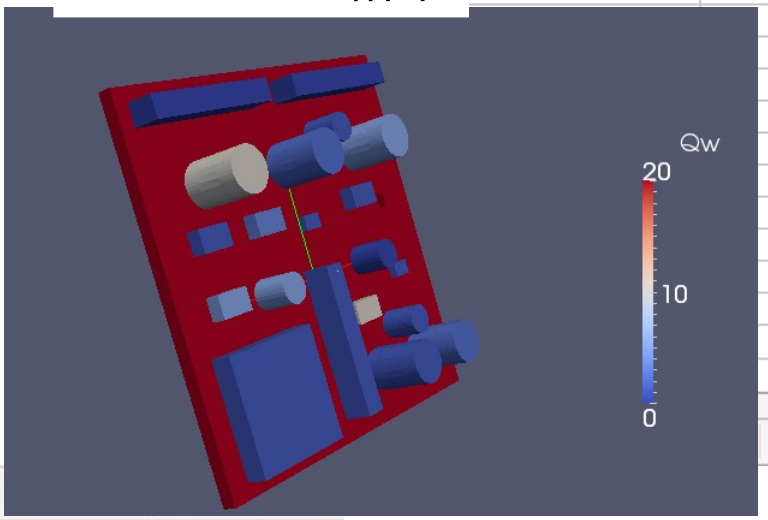
Liberation Sans 10

A1 <setFieldsDict>

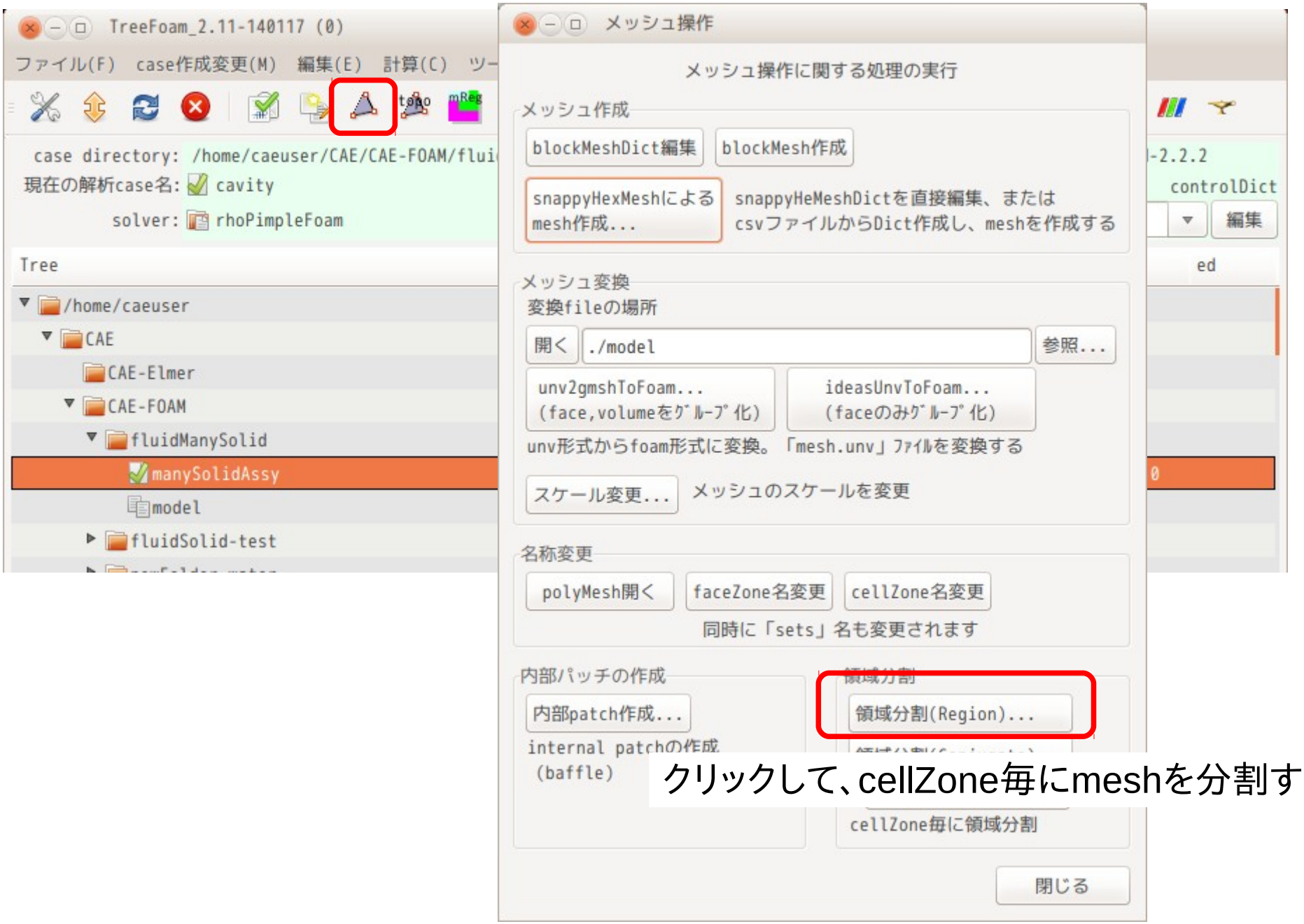
	A	B	C	D	E
1	<setFieldsDict>				
2	items	geometry data	Qw	T	
3	defaultFieldValues		0	300	
4	base		20		
5	cyli1		10		
6	cyli2		2		
7	cyli3		5		
8	cyli4		5		
9	cyli5		0		
10	cyli6		1		
11	cyli7		2		
12	cyli8		1		
13	cyli9		1		
14	heatSink		0		
15	rect1		0		
	rect10		1		
	rect11		1		
			5		
			2		
			0		
			1		
			3		
			1		
			1		
			1		
			20		
			10		

標準

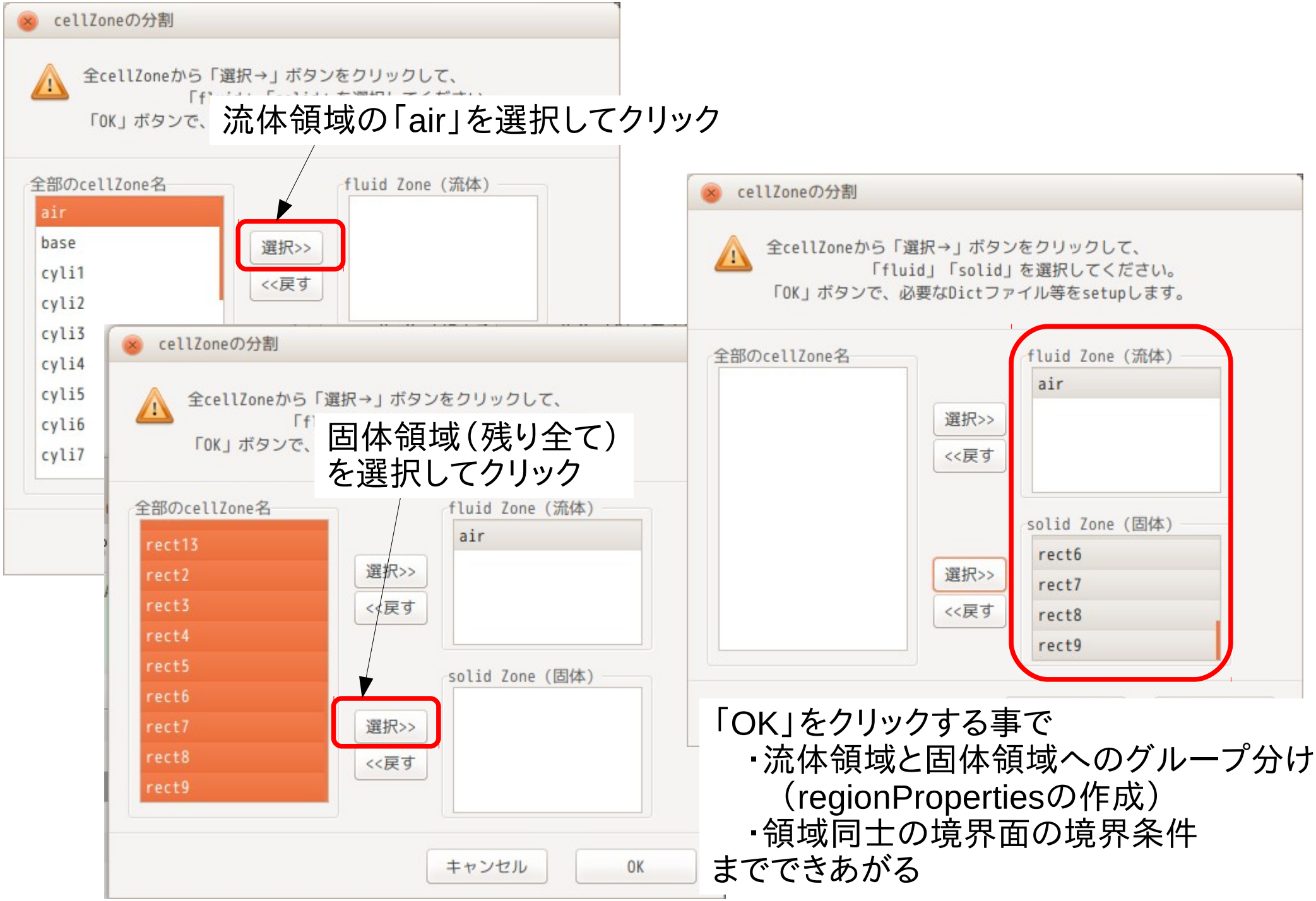
データセット結果



3.　region分割、計算開始



分割した領域を流体と固体にグループ分け



「OK」をクリックする事で
- 流体領域と固体領域へのグループ分け（regionPropertiesの作成）
- 領域同士の境界面の境界条件

までできあがる

できあがったregionProperties

regionProperties (~/CAE/CAE-FOAM/newFolder_0/manySolidAssy/regCase/constant

ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

開く 保存 元に戻す

regionProperties

```
/*---------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6.x                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.2;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      regionProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

regions
(
    fluid       ( air )
    solid       ( base cyli1 cyli2 cyli3 cyli4 cyli5 cyli6 cyli7 cyli8 cyli9
heatSink rect1 rect10 rect11 rect12 rect13 rect2 rect3 rect4 rect5 rect6 rect7
rect8 rect9 )
);

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

流体のairと固体のair以外にグループ分けできている

できあがった領域同士の境界面の境界条件

流体「air」の境界条件

ファイル(F)　編集(E)　表示(V)

	define patch at 0/air (boundary)	Qw	T	U
field type dimensions		volScalarField; [1 2 -3 0 0 0 0];	volScalarField; [0 0 0 1 0 0 0];	volVectorField; [0 1 -1 0 0 0 0];
internal Field		uniform 0;	uniform 300;	uniform (0 0 0);
caseWall	type wall;	type zeroGradient;	type zeroGradient;	type zeroGradient;
inW	type wall;	type zeroGradient;	type zeroGradient;	type zeroGradient;
outW	type wall;	type zeroGradient;	type zeroGradient;	type zeroGradient;
air_to_cyl i9	type mappedWall; sampleMode nearestPatchFace; sampleRegion cyli9; samplePatch cyli9_to_air; offsetMode uniform; offset (0 0 0);	type zeroGradient; value uniform 0;	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform 300; neighbourFieldName T; kappa fluidThermo; kappaName none;	type fixedValue; value uniform (0 0 0);
air_to_rec t9	type mappedWall; sampleMode nearestPatchFace; sampleRegion rect9; samplePatch rect9_to_air;	type zeroGradient; value uniform 0;	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform 300; neighbourFieldName T; kappa fluidThermo;	type fixedValue; value uniform (0 0 0);

T、Uの境界面の境界条件が設定できている。
（boundaryと各fieldのboundaryFieldの境界面）

流体「air」の境界条件の設定

boundaryFieldを設定
tutorialsのchtMultiRegionFoamを参考に設定

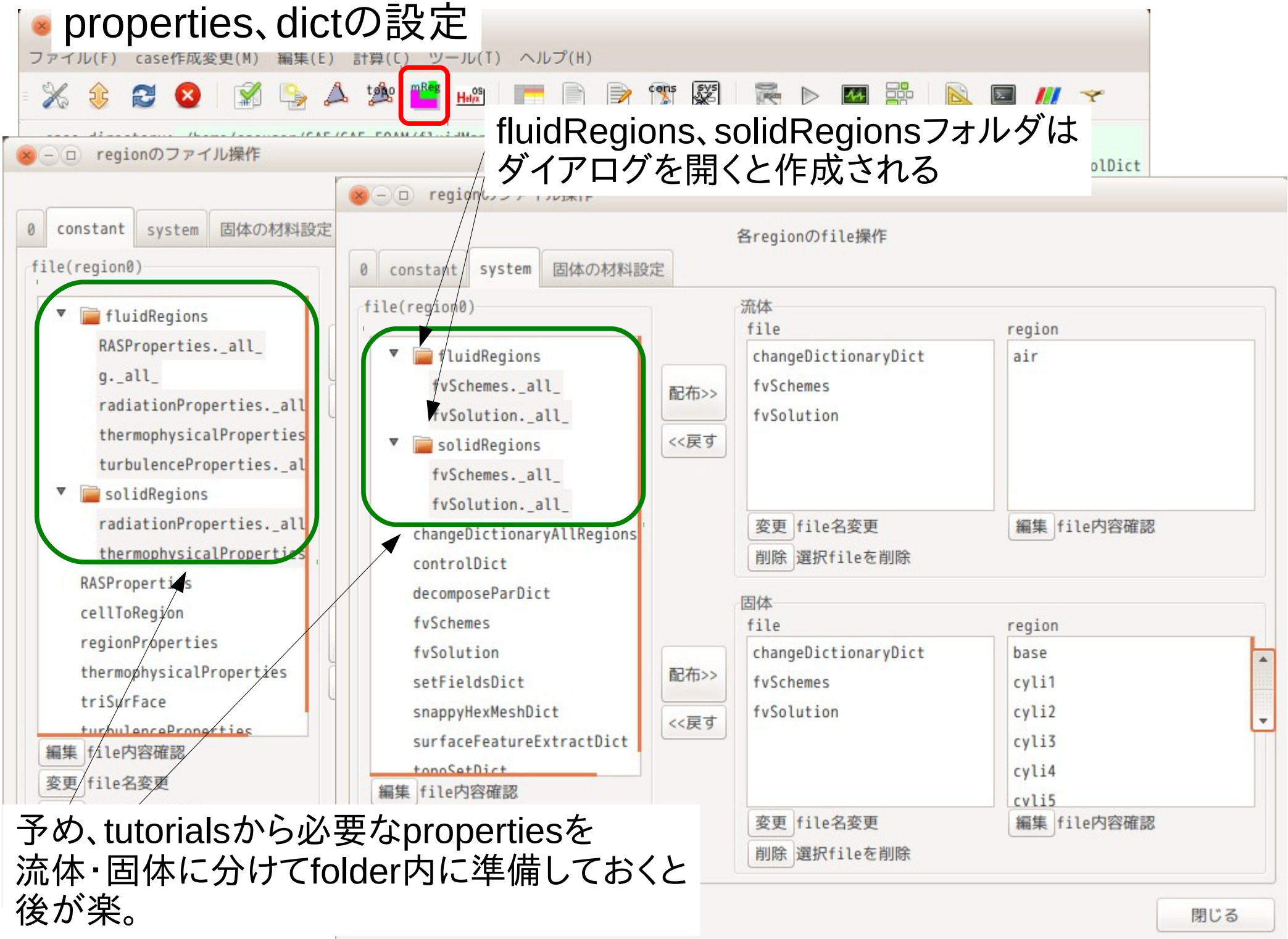
gridEditor : regcase/0/air (0.1)

ファイル(F)　編集(E)　表示(V)

	U	epsilon	k	p	p_rgh
field type dimensions	volVectorField; [0 1 -1 0 0 0 0];	volScalarField; [0 2 -3 0 0 0 0];	volScalarField; [0 2 -2 0 0 0 0];	volScalarField; [1 -1 -2 0 0 0 0];	volScalarField; [1 -1 -2 0 0 0 0];
internal Field	uniform (0 0 0);	uniform 0.01;	uniform 0.1;	uniform 100000;	uniform 100000;
caseWall	type fixedValue; value uniform (0 0 0);	type compressible::epsilonWallFunction; value uniform 0.01;	type compressible::kqRWallFunction; value uniform 0.1;	type calculated; value uniform 100000;	type fixedFluxPressure; value uniform 100000;
inW	type zeroGradient;	type compressible::epsilonWallFunction; value uniform 0.01;	type compressible::kqRWallFunction; value uniform 0.1;	type calculated; value uniform 100000;	type fixedFluxPressure; value uniform 100000;
outW	type fixedValue; value uniform (0 0 1);	type compressible::epsilonWallFunction; value uniform 0.01;	type compressible::kqRWallFunction; value uniform 0.1;	type calculated; value uniform 100000;	type fixedFluxPressure; value uniform 100000;
air_to_cyl i9	type fixedValue; value uniform (0 0 0);	type compressible::epsilonWallFunction; value uniform 0.01;	type compressible::kqRWallFunction; value uniform 0.1;	type calculated; value uniform 100000;	type fixedFluxPressure; value uniform 100000;
air_to_rec t9	type fixedValue; value uniform (0 0 0);	type compressible::epsilonWallFunction; value uniform 0.01;	type compressible::kqRWallFunction; value uniform 0.1;	type calculated; value uniform 100000;	type fixedFluxPressure; value uniform 100000;

排気部の為、
Z方向に流速を設定

properties、dictの設定

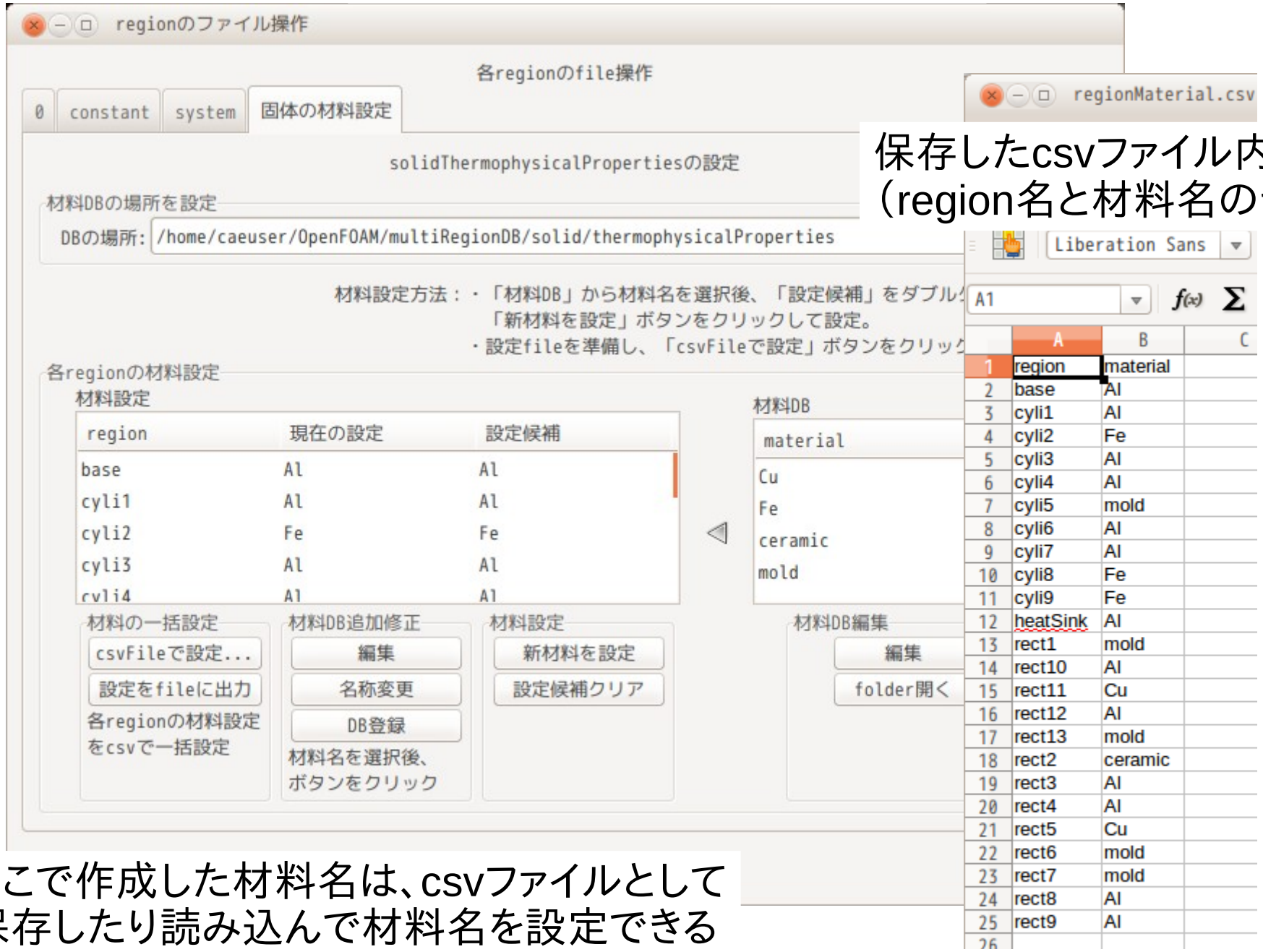


fluidRegions、solidRegionsフォルダは
ダイアログを開くと作成される

予め、tutorialsから必要なpropertiesを
流体・固体に分けてfolder内に準備しておくと
後が楽。

材料の設定

DBから材料名を選択して設定する



保存したcsvファイル内容
（region名と材料名のセット）

ここで作成した材料名は、csvファイルとして
保存したり読み込んで材料名を設定できる

solverの作成

chtMultiRegionFoamを発熱を考慮したsolver「chtQvMultiRegonFoam」に改造する。

createFluidFields.H

```
PtrList<fv::IOoptionList> fluidFvOptions(fluidRegions.size());

//---追加---
PtrList<volScalarField> QwFluid(fluidRegions.size());
//----------
```

追加

```
        i,
        new fv::IOoptionList(fluidRegions[i])
    );

    //追加----------------------------------------
    Info<< "    Adding to QwFluid\n" << endl;
    QwFluid.set
    (
        i,
        new volScalarField
        (
            IOobject
            (
                "Qw",
                runTime.timeName(),
                fluidRegions[i],
                IOobject::MUST_READ,
                IOobject::AUTO_WRITE
            ),
            fluidRegions[i]
        )
    );
    //-----------------------------------------
```

追加

createSolidFields.H

```
PtrList<volScalarField> betavSolid(solidRegions.size());

//---追加---
PtrList<volScalarField> QwSolid(solidRegions.size());
//----------
```

```
    }

    //追加----------------------------------------
    Info<< "    Adding to QwSolid\n" << endl;
    QwSolid.set
    (
        i,
        new volScalarField
        (
            IOobject
            (
                "Qw",
                runTime.timeName(),
                solidRegions[i],
                IOobject::MUST_READ,
                IOobject::AUTO_WRITE
            ),
            solidRegions[i]
        )
    );
    //-----------------------------------------
```

setRegionFluidFields.H

```
20
21    radiation::radiationModel& rad = radiation[i];
22
23    fv::IOoptionList& fvOptions = fluidFvOptions[i];
24
25    //追加--------------
26    volScalarField& Qw = QwFluid[i];
27    //------------------
28
29    const dimensionedScalar initialMass
```

追加

setRegionSolidFields.H

```
17
18    const volScalarField& betav = betavSolid[i];
19
20    fv::IOoptionList& fvOptions = solidHeatSources[i];
21
22    //追加--------------
23    volScalarField& Qw = QwSolid[i];
24    //------------------
25
```

追加

各regionで使用する変数名を定義

EEqn.H（fluid側）

```
2     volScalarField& he = thermo.he();
3
4     //追加-------------
5     dimensionedScalar partVol
6             ("partVol", dimVolume, gSum(mesh.V()));
7     volScalarField Qv = Qw / partVol;
8     //-----------------
9
10    fvScalarMatrix EEqn
11    (
12        fvm::ddt(rho, he) + fvm::div(phi, he)
13      + fvc::ddt(rho, K) + fvc::div(phi, K)
14      + (
15            he.name() == "e"
16          ? fvc::div
17            (
18                fvc::absolute(phi/fvc::interpolate(rho), U),
19                p,
20                "div(phiv,p)"
21            )
22          : -dpdt
23        )
24      - fvm::laplacian(turb.alphaEff(), he)
25     ==
26        rad.Sh(thermo)
27      + fvOptions(rho, he)
28      //追加--------
29      + Qv
30      //------------
```

追加

ワット数をregionの体積で除してW/m3に単位変換

追加

発熱分を方程式に追加

solveSolid.H（solid側）

```
{
    mesh.data::add("finalIteration", true);
}

{
    //追加--------------
    dimensionedScalar partVol
            ("partVol", dimVolume, gSum(mesh.V()));
    volScalarField Qv = Qw / partVol;
    //------------------

    for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
    {
        tmp<fvScalarMatrix> hEqn
        (
            fvm::ddt(betav*rho, h)
          - fvm::laplacian(betav*alpha, h, "laplacian(alpha,h)")
          ==
            fvOptions(rho, h)
          //追加--------
          + Qv
          //------------
        );
```

追加　ワット数をregionの体積で除してW/m3に単位変換

追加　発熱分を方程式に追加

ソースコードを修正後、solver名を「cthQvMultiRegionFoam」としてコンパイルする。

計算開始

controlDictを修正して実行する。

```
2 | =========                 |
3 | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
4 |  \\    /   O peration     | Version:  2.2.2
5 |   \\  /    A nd           | Web:      www.OpenFOAM.org
6 |    \\/     M anipulation  |
7 \*---------------------------------------------------------------------------
8 FoamFile
9 {
10     version     2.0;
11     format      ascii;
12     class       dictionary;
13     location    "system";
14     object      controlDict;
15 }
16 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
17
18 libs
19 (
20     "libcompressibleTurbulenceModel.so"
21     "libcompressibleRASModels.so"
22 );
23
24 //application     chtMultiRegionFoam;
25 application     chtQvMultiRegionFoam;
26
27 startFrom       latestTime;
28
29 startTime       0.001;
30
31 stopAt          endTime;
32
33 endTime         10;
```

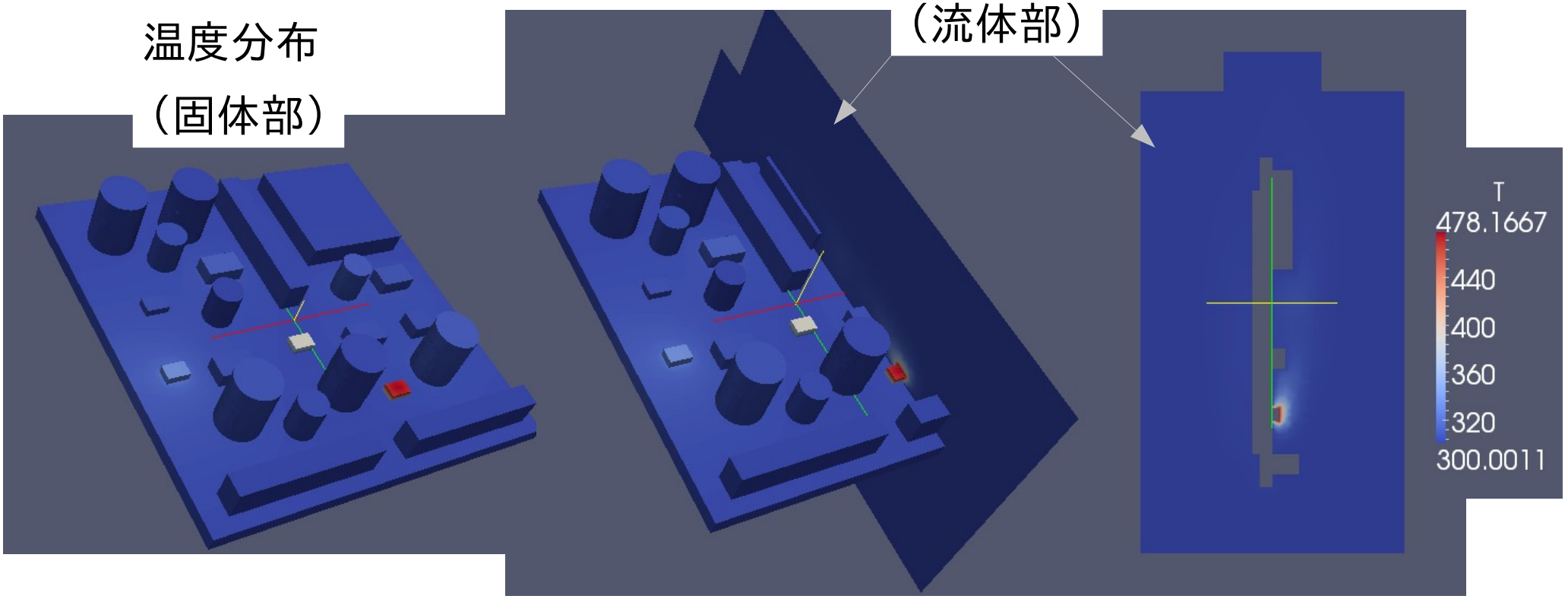
lib名を記述
（tutorialsと同じ）

solver名を書き換え
（chtQvMultiRegionFoam）

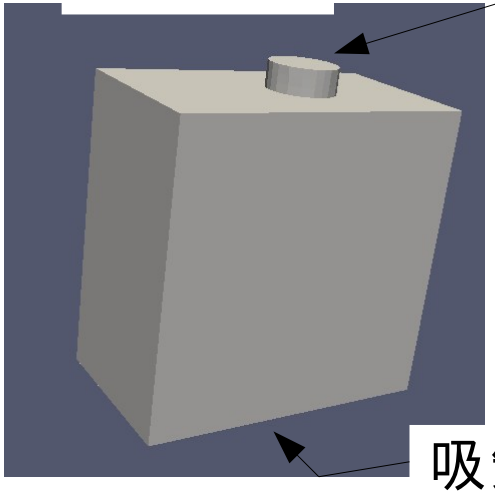
温度分布

（固体部）

（流体部）



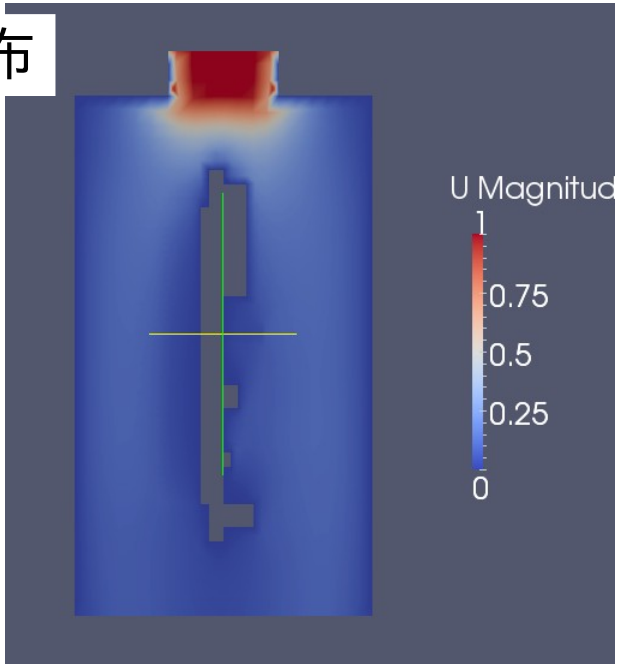
全体形状



排気面
（1m/s）

吸気面
（底面）

速度分布



入力データ

stlファイルと以下のcsvデータが準備できればデータセットできる。

	A	B	C	D	E	F	G	H	I
1									
2	<blockMesh>		x	y	z				
3		cellSize	0.002						
4		overBlockSize	5						
5									
6	<snappyHexMesh>								
7		mesh	0.025	0	0.025				
8	stlFile	sect (patch/cellZone/reg)	featureEdge cellSize	base cellSize	fine cellSize		Qw		material
9	caseWall	patch	0.002	0.002					
10	inW	patch	0.002	0.002					
11	outW	patch	0.002	0.002	0.001				
12	base	cellZone	0.001	0.001			20		Al
13	cyli1	cellZone	0.001	0.001	0.0005		10		Al
14	cyli2	cellZone	0.001	0.001	0.0005		2		Fe
15	cyli3	cellZone	0.001	0.001	0.0005		5		Al
16	cyli4	cellZone	0.001	0.001	0.0005		5		Al
17	cyli5	cellZone	0.001	0.001	0.0005		0		mold
18	cyli6	cellZone	0.001	0.001	0.0005		1		Al
19	cyli7	cellZone	0.001	0.001	0.0005		2		Al
20	cyli8	cellZone	0.001	0.001	0.0005		1		Fe
21	cyli9	cellZone	0.001	0.001	0.0005		1		Fe
22	heatSink	cellZone	0.001	0.001	0.00025		0		Al
23	rect1	cellZone	0.001	0.001			0		mold
24	rect10	cellZone	0.001	0.001	0.0005		1		Al
25	rect11	cellZone	0.001	0.001			1		Cu

メッシュ作成データ

入力データ
（ワット数）

固体の物性値

4.　まとめ

chtMultiRegionFoam（固体・流体熱連成）を
本格的に使いこなす為には、根気と労力が必要。
領域が多くあると、不可能に近い。

今回、TreeFoam上でメッシュ作成とデータセットがcsvデータを
準備する事で可能に。

TreeFoam上で、メッシュ作成、setFields、領域分割、
領域間の境界条件、固体の物性値設定ができるので、
領域数が増えても、負担はそれほど増えず。
（単純作業は、殆どTreeFoamがやってくれる）

TreeFoamの英語版も完成。（DEXCS用も）
　TreeFoam-2.11-140208.tar.gz