

TreeFoamによる

multiRegionモデルの境界条件の自動設定と  
field内変数の解釈方法について

14/11/1  
藤井

# TreeFoamによるmultiRegionモデルの作成方法

メッシュ作成	snappyHexMesh	リスト形式データで処理
↓		
fieldにデータセット	setFields	リスト形式データで処理
↓		
領域分割	splitMesh	GUIで処理
↓		
流体・固体領域に ファイルセット	fvSolution、 各Properties 等	GUIで処理
↓		
固体の材料セット	thermophysicalProperties	リスト形式データで処理
↓		
境界条件セット	各region毎に設定	gridEditorで処理

領域分割した段階で、領域間の境界条件は、  
T、U fieldのみpatch内容が設定される。  
他のepsilon, k, p, p\_rghは、zeroGradientの設定なので、  
再設定する必要がある。

→領域分割時に境界条件を設定

流体・固体領域専用folderを準備しているが、  
この中のfileを手動でコピーしている

→領域分割時にコピー

# 領域分割後の境界条件 ＜現状の状態＞

	T	U	epsilon	k
inW	type fixedValue; value uniform 300;	type zeroGradient;	type fixedValue; value uniform 0.01;	type inletOutlet; inletValue uniform 0.1; value uniform 0.1;
outW	type zeroGradient;	type fixedValue; value uniform ( 0 0 0.1 );	type zeroGradient;	type zeroGradient;
air_to_cyl i9	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform 300; neighbourFieldName T; K basicThermo; KName none;	type fixedValue; value uniform ( 0 0 0 );	type zeroGradient; value uniform 0;	type zeroGradient; value uniform 0;
air_to_rec t9	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform 300; neighbourFieldName T; K basicThermo; KName none;	type fixedValue; value uniform ( 0 0 0 );	type zeroGradient; value uniform 0;	type zeroGradient; value uniform 0;
air_to_rec t7	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform 300; neighbourFieldName T; K basicThermo; KName none;	type fixedValue; value uniform ( 0 0 0 );	type zeroGradient; value uniform 0;	type zeroGradient; value uniform 0;

T、U は、設定されている

T、U 以外は、「zeroGradient」  
再設定する必要がある。

# 領域分割後の境界条件 ＜修正後＞

gridEditor: regCase/0/air (0:1)				
ファイル(F) 編集(E) 表示(V)				
	T	U	epsilon	k
field type	volScalarField;	volVectorField;	volScalarField;	volScalarField;
dimensions	[ 0 0 0 1 0 0 0 ];	[ 0 1 -1 0 0 0 0 ];	[ 0 2 -3 0 0 0 0 ];	[ 0 2 -3 0 0 0 0 ];
otherNames	iniTemp 300; iniVelocity (0 0 0); zeroVelocity (0 0 0); iniPress 100000; epsilon 0.01; k 0.1;	iniTemp 300; iniVelocity (0 0 0); zeroVelocity (0 0 0); iniPress 100000; epsilon 0.01; k 0.1;	iniTemp 300; iniVelocity (0 0 0); zeroVelocity (0 0 0); iniPress 100000; epsilon 0.01; k 0.1;	iniTemp 300; iniVelocity (0 0 0); zeroVelocity (0 0 0); iniPress 100000; epsilon 0.01; k 0.1;
internal Field	uniform 300;	uniform \$iniVelocity;	uniform \$epsilon;	uniform \$k;
caseWall	type zeroGradient;	type fixedValue; value uniform ( 0 0 0 );	type compressible::epsilonWallFunction; value uniform 0.01; Cmu 0.09; kappa 0.41; E 9.8; value uniform 0.01;	type compressible::kqRWallF value uniform 0.1;
inW	type fixedValue; value uniform 300;	type zeroGradient;	type fixedValue; value uniform 0.01;	type inletOutlet; inletValue uniform 0.1; value uniform 0.1;
outW	type zeroGradient;	type fixedValue; value uniform ( 0 0 0.1 );	type zeroGradient;	type zeroGradient;
	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform \$iniTemp;	type fixedValue; value uniform \$zeroVelocity;	type compressible::epsilonWallFunction; value uniform \$epsilon;	type compressible::kqRWallF value uniform \$k;

- ・変数が設定されている
- ・internalFieldが変数として設定されている

# 領域分割後の境界条件 ＜修正後＞

gridEditor: regCase/0/air (0:1)				
ファイル(F) 編集(E) 表示(V)				
	T	U	epsilon	k
inW	type fixedValue; value uniform 300;	type zeroGradient;	type fixedValue; value uniform 0.01;	type inletOutlet; inletValue uniform 0.1; value uniform 0.1;
outW	type zeroGradient;	type fixedValue; value uniform ( 0 0 0.1 );	type zeroGradient;	type zeroGradient;
air_to_cyl i9	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform \$iniTemp; Tnbr T; kappa fluidThermo; kappaName none;	type fixedValue; value uniform \$zeroVelocity;	type compressible::epsilonWallFunction; value uniform \$epsilon;	type compressible::kqRWallF value uniform \$k;
air_to_rec t9	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform \$iniTemp; Tnbr T; kappa fluidThermo; kappaName none;	type fixedValue; value uniform \$zeroVelocity;	type compressible::epsilonWallFunction; value uniform \$epsilon;	type compressible::kqRWallF value uniform \$k;
air_to_rec t7	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform \$iniTemp; Tnbr T; kappa fluidThermo; kappaName none;	type fixedValue; value uniform \$zeroVelocity;	type compressible::epsilonWallFunction; value uniform \$epsilon;	type compressible::kqRWallF value uniform \$k;
air_to_cyl	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform \$iniTemp; Tnbr T;	type fixedValue; value uniform \$zeroVelocity;	type compressible::epsilonWallFunction; value uniform \$epsilon;	type compressible::kqRWallF value uniform \$k;

領域間の境界条件が全て設定されている。

# fieldの内容

```
U (~/TreeFoam/temp/U.0) - gedit
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
U x
16 // ****
17
18 dimensions      [ 0 1 -1 0 0 0 ];
19 #include "${FOAM_CASE}/constant/include/variableSetting"
20 #include "${FOAM_CASE}/constant/include/boundaryConditionsFluid"
21
22 internalField    uniform $iniVelocity;
23
24 boundaryField
25 {
26     caseWall
27     {
28         type      fixedValue;
29         value      uniform ( 0 0 0 );
30     }
31     inW
32     {
33         type      zeroGradient;
34     }
35     outW
36     {
37         type      fixedValue;
38         value      uniform ( 0 0 0.1 );
39     }
40     ".*_to_.*"
41     {
42         $:wallToRegion.U;
43     }
44 }
45
46
47 // ****
```

includeFileを設定

変数「\$iniVelocity」でinternalFieldの値を設定

wildCard「.\*\_to\_.\*」を使ってregion間のpatch名を指定。  
変数「\$:wallToRegion.U」を使ってpatch内容を指定。



# includeFileの内容

## <boundaryConditionsFluidの内容>

```
13 location    "";
14 object      regionWallBoundaryConditionsDict;
15 }
16 // *****
17
18 wallToRegion
19 {
20
21     T
22     {
23         type compressible::turbulentTemperatureCoupledBaffleMixed;
24         value uniform $iniTemp;
25         Tnbr T;
26         kappa fluidThermo;
27         kappaName none;
28     }
29
30     U
31     {
32         type fixedValue;
33         value uniform $zeroVelocity;
34     }
35
36     epsilon
37     {
38         type compressible::epsilonWallFunction;
39         value uniform $epsilon;
40     }
41
42     k
43     {
44         type compressible::kqRWallFunction;
```

変数「\$zeroVelocity」を使用

変数「\$.wallToRegion.U」の内容

## <variableSettingの内容>

```
1 /*-----* C++ *-----
2 | =====
3 | \ \      /  F i e l d      | OpenFOAM: The Open Sou
4 | \ \      /  O p e r a t i o n  | Version:  2.3.0
5 |  \ \      /  A n d          | Web:      www.OpenFOAM
6 |  \ \      /  M a n i p u l a t i o n  |
7 |-----*
8 FoamFile
9 {
10     version      2.1;
11     format        ascii;
12     class         dictionary;
13     location      "";
14     object        regionWallBoundaryConditionsDict;
15 }
16 // *****
17
18
19     iniTemp      300;
20     iniVelocity   (0 0 0);
21     zeroVelocity  (0 0 0);
22     iniPress      100000;
23     epsilon       0.01;
24     k             0.1;
```

変数定義の内容

# includeFileと各fieldの作成方法

「regionWallBoundaryConditionsDict」により指示して作成する

## regionWallBoundaryConditionsDictの内容

```
regionWallBoundaryConditionsDict
12 class dictionary,
13 location "";
14 object regionWallBoundaryConditionsDict;
15 }
16 // ****
17
18 // directory of include files
19 // below include files are created at next below directory.
20 // - variableSetting
21 // - boundaryConditionsFluid
22 // - boundaryConditionsSolid
23 //
24 includeDir "${FOAM_CASE}/include";
25
26
27 // variable setting
28 variableSetting
29 {
30     iniTemp      300;
31     iniVelocity  (0 0 0);
32     zeroVelocity (0 0 0);
33     iniPress     100000;
34     turbEpsilon  0.01;
35     turbK        0.1;
36 }
37
38
```

includeDirを指定

変数を定義

```
regionWallBoundaryConditionsDict
36 }
37
38
39 //internal fields setting for variableName
40 // If field does not exist in region, is not apply, so is not change
41 // And if internalField type is nonuniform, should not be applied.
42 //
43 // example:
44 //     U      iniVelocity;
45 //
46 // InternalField of U is set 'internalField uniform $iniVelocity;'.
47 // but if its internalField type is nonuniform, should not be applied.
48 //
49 internalFields
50 {
51     fluidRegions
52     {
53         U      iniVelocity;
54         epsilon  turbEpsilon;
55         k      turbK;
56         p      iniPress;
57         p_rgh  iniPress;
58     }
59     solidRegions
60     {
61         p      iniPress;
62     }
63 }
64
65
```

internalFieldの設定  
変数で設定する  
ここにリストアップ



```

regionWallBoundaryConditionsDict x
65
66 //boundary conditions of walls between regions
67 // If the field does not exist in region, its boundary condition is
68 //
69 regionWallBoundaryConditions
70 {
71     fluidRegions
72     {
73         T
74         {
75             type compressible::turbulentTemperatureCoupledBaffleMixed
76             value uniform $iniTemp;
77             Tnbr T;
78             kappa fluidThermo;
79             kappaName none;
80         }
81         U
82         {
83             type fixedValue;
84             value uniform $zeroVelocity;
85         }
86     }
87     epsilon
88     {
89         type compressible::epsilonWallFunction;
90         value uniform $turbEpsilon;
91     }
92 }
93
94 k
--

```

## 境界条件の設定

### 流体の境界条件

```

regionWallBoundaryConditionsDict x
102     type calculated;
103     value uniform $iniPress;
104 }
105
106     p_rgh
107     {
108         type fixedFluxPressure;
109         value uniform $iniPress;
110         gradient uniform 0;
111     }
112 }
113
114     solidRegions
115     {
116         T
117         {
118             type compressible::turbulentTemperatureCoupledBaffleMixed;
119             value uniform $iniTemp;
120             Tnbr T;
121             kappa solidThermo;
122             kappaName none;
123         }
124
125         p
126         {
127             type calculated;
128             value uniform $iniPress;
129         }
130     }
131 }
132 }
133
134 // *****

```

### 固体の境界条件

領域分割前に、system/regionWallBoundaryConditionsDictを準備することで、Dict内容に従って、includeファイルを作成し、境界条件を設定する。

# TreeFoam上の操作

TreeFoam\_2.24-141012 (0)

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

case directory: /home/caeuser/CAE/CAE-FOAM/OF-2.3/coolingModel  
現在の解析case名: ☒ chtMultiRegionCase\_copy1\_copy0  
solver: ☒ chtMultiRegionFoam

Tree

- coolingModel
  - chtMultiRegionCase
    - chtMultiRegionCase\_copy0
    - chtMultiRegionCase\_copy0\_copy0
    - chtMultiRegionCase\_copy1
    - ☒ chtMultiRegionCase\_copy1\_copy0

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

TreeFoam ver 2.24-141012 (0) を起動しました。  
OpenFOAM - 2.3.0

合計 32.62 GB, 空き 2.5 GB

**クリック**

**caseを選択  
(領域分割直前の状態)**

**メッシュ操作**

メッシュ操作に関する処理の実行

メッシュ作成

blockMeshDict編集 blockMesh作成

snappyHexMeshによる mesh作成... snappyHexMeshDictを直接編集、または csvファイルからDict作成し、meshを作成する

メッシュ変換

変換fileの場所

開く ./model 参照...

unv2gmshToFoam... (face,volumeをグループ化) ideasUnvToFoam... (faceのみグループ化)

unv形式からfoam形式に変換。「mesh.unv」ファイルを変換する

スケール変更... メッシュのスケールを変更

名称変更

polyMesh開く faceZone名変更 cellZone名変更

同時に「sets」名も変更されます

内部パッチの作成

内部patch作成... internal patchの作成 (baffle)

**領域分割**

**領域分割(Region)...**

**クリックして  
領域分割開始**

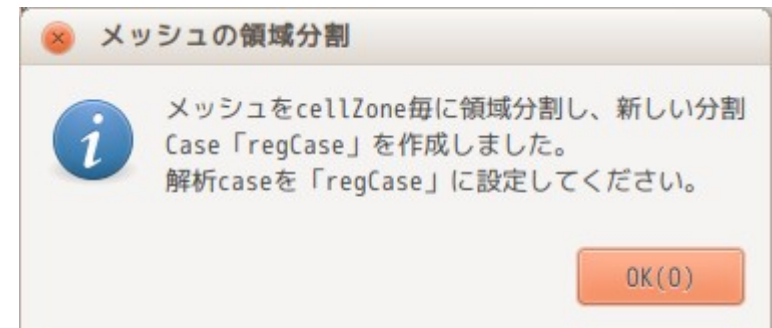
閉じる



この設定により、「regionProperties」が作成され、新しいcase「regCase」が出来上がる。

領域分割時、「regionWallBoundaryConditionsDict」の設定に従って、includeファイルを作成し、境界条件を設定する。

case内に「regionWallBoundaryConditionsDict」が存在しなかった場合は、TreeFoam/data/multiRegionDBフォルダ内からOFバージョンに合うDictをコピーする



# 領域分割後の状態

gridEditor: regCase/0/air (0:1)

ファイル(F)

編集(E)

表示(V)

	define patch (boundary)	T	epsilon
field type		volScalarField;	
dimensions		[ 0 0 0 1 0 0 0 ];	
otherNames		iniTemp 300; iniVelocity (0 0 0); zeroVelocity (0 0 0); iniPress 100000; turbEpsilon 0.01; turbK 0.1;	
internal Field		uniform 300;	
caseWall	type wall; inGroups 1(wall);	type zeroGradient;	
inW	type wall; inGroups 1(wall);	type fixedValue; value uniform 300;	
outW	type wall; inGroups 1(wall);	type zeroGradient;	
air_to_cyl i9	type mappedWall; inGroups 1(wall); sampleMode nearestPatchFace; sampleRegion cyli9; samplePatch cyli9_to_air;	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform \$iniTemp; Tnbr T; kappa fluidThermo; kappaName none;	
	type mappedWall; inGroups 1(wall);	type compressible::turbulentTemperatureCoupledBaffleMixed; value uniform \$iniTemp;	

「T」fieldの内容

```

14
15 }
16 // *****
17
18 dimensions      [ 0 0 0 1 0 0 0 ];
19 #include "${FOAM_CASE}/include/variableSetting"
20 #include "${FOAM_CASE}/include/boundaryConditionsFluid"
21
22 internalField    uniform 300;
23
24 boundaryField
25 {
26     caseWall
27     {
28         type          zeroGradient;
29     }
30     inW
31     {
32         type          fixedValue;
33         value         uniform 300;
34     }
35     outW
36     {
37         type          zeroGradient;
38     }
39     ".*_to_.*"
40     {
41         $:wallToRegion.T;
42     }
43 }
44
45
46 // *****

```

region間の境界条件は、設定できている。



# 領域分割後に、境界条件を修正する場合 (Dictを編集する場合)

TreeFoam\_2.24-141012 (0)

ファイル(F) case作成変更(M) 編集(E) ヘルプ(H)

case directory: ...E-FOAM/OF-2.3/coolingModel/chtMultiRegionFoam

現在の解析case名: ☒ regCase

solver: ☒ chtMultiRegionFoam

Tree

- ▶ chtMultiRegionCase\_copy0\_copy0
- ▶ chtMultiRegionCase\_copy1
- ▼ chtMultiRegionCase\_copy1\_copy0
  - model
  - ▶ ☒ regCase
- ▶ newFolder\_1

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

合計 32.62 GB, 空き 2.46 GB

クリック

クリック(新しく追加)

境界条件の編集・設定

各regionの境界面の境界条件を編集・設定する。  
includeファイルや変数の設定を行う。

changeDictionaryAllRegionsの設定  
全regionに対して一括して変更を加える。

boundaryと各fieldの設定を読み込み、  
changeDictionaryAllRegionsDictを作成する。

changeDictionaryAllRegionsDictを編集・実行する。  
この実行により全regionの設定が変更される。

region内のfile操作

設定されたregion名を変更・追加・削除する。  
regionを追加した場合、meshをコピーする必要がある。

各regionへのfileコピーや固体領域の材料設定を行う。

## regionBoundaryEditorが起動する

境界条件の設定

領域間の境界条件の編集・設定 (Region Boundary Editor)

変数設定 (fluid solid 共用)

変数名、値の定義

iniTemp	300;
iniVelocity	(0 0 0);
zeroVelocity	(0 0 0);
iniPress	100000;
turbEpsilon	0.01;
turbK	0.1;

fluid, solid領域で使用する変数を定義。

この画面上で、  
regionWallBondaryConditionsDictの内容を修正し、  
internalField、境界条件を再設定することができる。

fieldの設定

field名を選択後、そのfieldの内容を設定する。  
internalFieldと境界面のboundaryFieldを設定する。

fluid領域

field名

internalField: (変更せず)

境界面の境界条件

```
type compressible::turbulentTemperatureCoupledBaffleMixed;  
value uniform $iniTemp;  
Tnbr T;  
kappa fluidThermo;  
kappaName none;
```

solid領域

field名

internalField: (変更せず)

境界面の境界条件

```
type compressible::turbulentTemperatureCoupledBaffleMixed;  
value uniform $iniTemp;  
Tnbr T;  
kappa solidThermo;  
kappaName none;
```

キャンセル 設定



# multiRegionの境界条件について

OFのversionで境界条件が変わってきている。

このたびに、プログラムを書き換え対応し、ver1.6から2.3まで対応してきた。

OF1.7  
以前

```
type                solidWallMixedTemperatureCoupled;  
neighbourFieldName  T;  
K                  K;  
value              uniform 0;
```

OF2.0

```
type                compressible::turbulentTemperatureCoupledBaffleMixed;  
value              uniform 300;  
neighbourFieldName T;  
K                  basicThermo;  
KName              none;
```

OF2.2

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

OF2.3

```
type                compressible::turbulentTemperatureCoupledBaffleMixed;  
value              uniform 300;  
Tnbr               T;  
kappa              fluidThermo;  
kappaName          none;
```

今回、regionWallBoundaryConditionsDictを作り、これで全てのregion間の境界条件を設定できる様に修正した。  
(今後は、Dictファイルを準備するだけ、プログラムの変更はしなくて済む。)

現状、OF2.2と2.3用のDictファイルを作っている。  
現在のOFのversionに応じてdefaultのDictファイルがコピーされる。

```
TreeFoam
  data
    multiRegionDB
      regionWallBoundaryConditionsDict_2.2.0
      regionWallBoundaryConditionsDict_2.3.0
```

OF2.1以前は、「" .\*" 」や「\$:wallToRegion.T」が使えないので、未対応。

# field内変数、patch名について

tutorialsの「cavity」で確認

## ＜boundaryの内容＞

```
17
18 3
19 (
20     movingWall
21     {
22         type            wall;
23         inGroups       1(wall);
24         nFaces          20;
25         startFace       760;
26     }
27     fixedWalls
28     {
29         type            wall;
30         inGroups        1(wall);
31         nFaces          60;
32         startFace       780;
33     }
34     frontAndBack
35     {
36         type            empty;
37         inGroups        1(empty);
38         nFaces          800;
39         startFace       840;
40     }
41 }
```

## ＜U fieldの内容＞

```
19 internalField    uniform (0 0 0);
20
21 boundaryField
22 {
23     /*movingWall
24     {
25         type        fixedValue;
26         value        uniform (1 0 0);
27     }*/
28     ".*"
29     {
30         type        fixedValue;
31         value        uniform (3 0 0);
32     }
33     wall
34     {
35         type        fixedValue;
36         value        uniform (2 0 0);
37     }
38     fixedWalls
39     {
40         type        fixedValue;
41         value        uniform (0 0 0);
42     }
43     frontAndBack
44     {
45         type        empty;
46     }
47 }
```

wildCardで設定

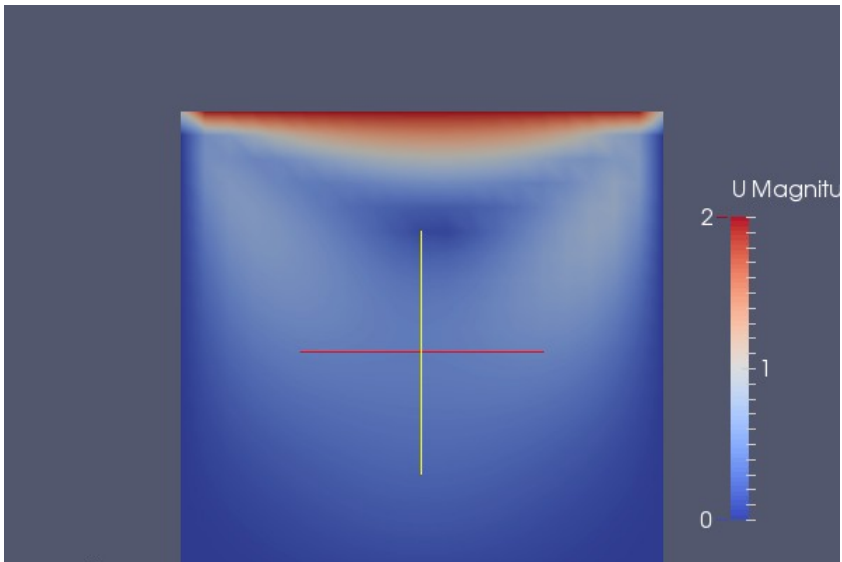
inGroupsで設定

movingWallには(2 0 0)が設定

```

19 internalField    uniform (0 0 0);
20
21 boundaryField
22 {
23     /*movingWall
24     {
25         type        fixedValue;
26         value        uniform (1 0 0);
27     }*/
28     fixedWalls
29     {
30         type        fixedValue;
31         value        uniform (0 0 0);
32     }
33     frontAndBack
34     {
35         type        empty;
36     }
37     wall
38     {
39         type        fixedValue;
40         value        uniform (2 0 0);
41     }
42     ".*"
43     {
44         type        fixedValue;
45         value        uniform (3 0 0);
46     }
47 }

```



	define patch at constant/. (boundary)	U	p
field type dimensions		volVectorField; [0 1 -1 0 0 0 0];	volScalarField; [0 2 -2 0 0 0 0];
internal Field		uniform (0 0 0);	uniform 0;
movingWall	type wall; inGroups 1(wall);	type fixedValue; value uniform (2 0 0);	type zeroGradient;
fixedWalls	type wall; inGroups 1(wall);	type fixedValue; value uniform (0 0 0);	type zeroGradient;
frontAndBa ck	type empty; inGroups 1(empty);	type empty;	type empty;

movingWallには(2 0 0)が設定

patch内容の設定には、優先順位がある。

1. 直接patch名を指定
2. inGroupsで指定
3. waildCardで指定

# patchの上書きについて

```
17
18 3
19 (
20   movingWall
21   {
22       type      wall;
23       inGroups   2(wall wallM);
24       nFaces    20;
25       startFace 760;
26   }
```

inGroupsを2ヶ設定

```
27   fixedWalls
28   {
29       type      wall;
30       inGroups   1(wall);
31       nFaces    60;
32       startFace 780;
33   }
34   frontAndBack
35   {
36       type      empty;
37       inGroups   1(empty);
38       nFaces    800;
39       startFace 0;
40   }
```

inGroupsが優先するので、  
(2.5 0 0) が設定される

```
27   }
28   fixedWalls
29   {
30       type      fixedValue;
31       value      uniform (0 0 0);
32   }
33   frontAndBack
34   {
35       type      empty;
```

inGroups

```
37   wall
38   {
39       type      fixedValue;
40       value      uniform (2 0 0);
41   }
42   wallM
43   {
44       type      fixedValue;
45       value      uniform (2.5 0 0);
46   }
```

この内容に上書き

wildCard

```
47   ".*"
48   {
49       type      fixedValue;
50       value      uniform (3 0 0);
51   }
52   "moving.*"
53   {
54       type      fixedValue;
55       value      uniform (3.5 0 0);
56   }
```

この内容に上書き

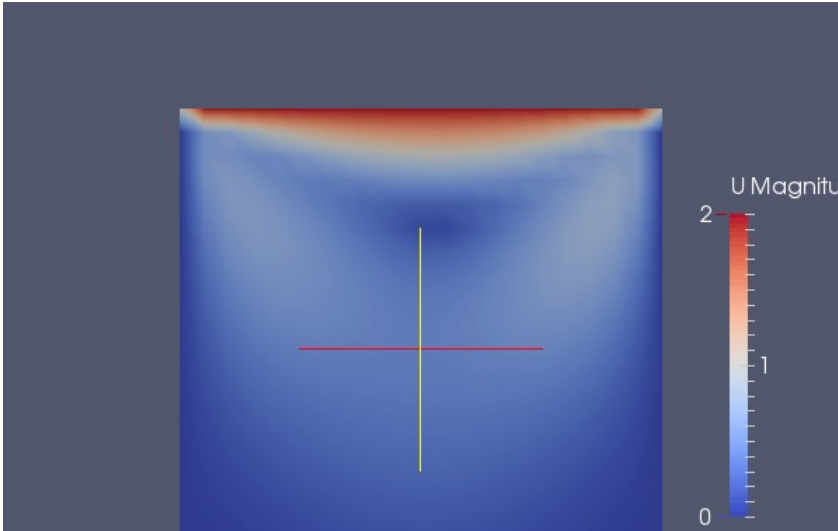


```

18
19 internalField    uniform (0 0 0);
20
21 boundaryField
22 {
23     movingWall
24     {
25         type      fixedValue;
26         value      uniform (1 0 0);
27     }
28     fixedWalls
29     {
30         type      fixedValue;
31         value      uniform (0 0 0);
32     }
33     frontAndBack
34     {
35         type
36     }
37     movingWall
38     {
39         type      fixedValue;
40         value      uniform (2 0 0);
41     }
42 }
43
44 // *****

```

movingWallを再設定



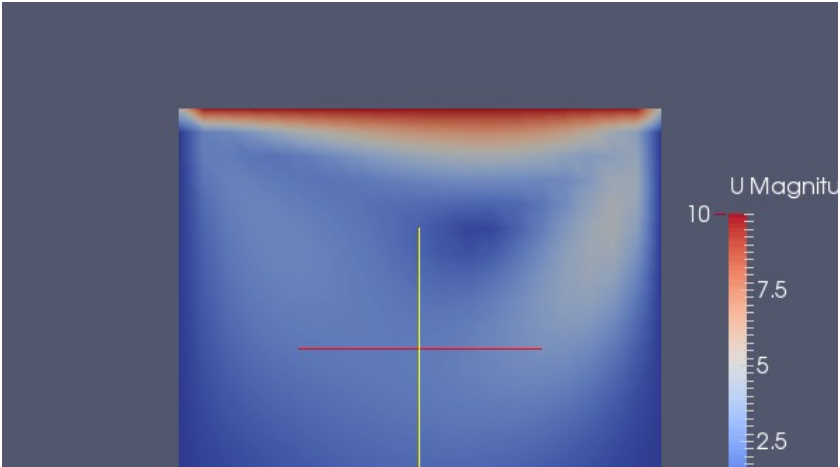
	define patch at constant/. (boundary)	U	p
field type dimensions		volVectorField; [0 1 -1 0 0 0 0];	volScalarField; [0 2 -2 0 0 0 0];
internal Field		uniform (0 0 0);	uniform 0;
movingWall	type wall; inGroups 2(wall wallM);	type fixedValue; value uniform (2 0 0);	type zeroGradient;
fixedWalls	type wall; inGroups 1(wall);	type fixedValue; value uniform (0 0 0);	type zeroGradient;
frontAndBa ck	type empty; inGroups 1(empty);	type empty;	type empty;

MovingWallが上書きされ(2 0 0)が設定される



# field内変数について

```
18
19 highVelocity      (10 0 0);
20 highVelocityField  uniform $highVelocity;
21
22 variableType
23 {
24     high
25     {
26         type      fixedValue;
27         value      $highVelocityField;
28     }
29 }
30
31 internalField      uniform (0 0 0);
32
33 boundaryField
34 {
35     constVelocity
36     { $:variableType.high; }
37     movingWall
38     { $constVelocity; }
39     fixedWalls
40     {
41         type      fixedValue;
42         value      uniform (0 0 0);
43     }
44     frontAndBack
45     {
46         type      empty;
47     }
48 }
49
```



	define patch at constant/. (boundary)	U	
field type dimensions		volVectorField; [0 1 -1 0 0 0 0];	volScal
otherNames		highVelocity (10 0 0); highVelocityField uniform \$highVelocity;	
internal Field		uniform (0 0 0);	uniform
movingWall	type wall; inGroups 2(wall wallM);	type fixedValue; value \$highVelocityField;	type 2
fixedWalls	type wall; inGroups 1(wall);	type fixedValue; value uniform (0 0 0);	type 2
frontAndBa ck	type empty; inGroups 1(empty);	type empty;	type e

MovingWallには、最終的に(10 0 0)が設定される。  
gridEditorでも上記のように解釈する。

「\$:variableType.hig」の様な階層型変数のアクセス先は  
boundaryFieldの外なので、外側に定義する。

```

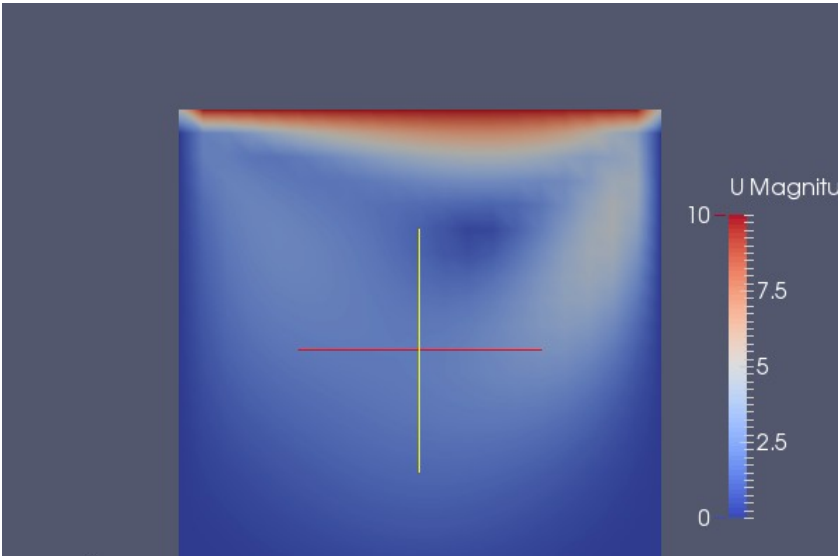
18
19 highVelocity      (10 0 0);
20 highVelocityField uniform $highVelocity;
21
22 variableType
23 {
24     high
25     {
26         type    fixedValue;
27         value    $highVelocityField;
28     }
29 }
30
31 internalField      uniform (0 0 0);
32
33 boundaryField
34 {
35     constVelocity
36     { $:variableType.high;    }
37     movingWall
38     { $constVelocity;        }
39     fixedWalls
40     {
41         type        fixedValue;
42         value        uniform (0 0 0);
43     }
44     frontAndBack
45     {
46         type        empty;
47     }
48 }
49

```

このタイプは、boundaryFieldの内外両方に定義できる

このタイプ(階層型変数)の変数定義は、boundaryFieldの外側にしか定義できない。

```
14 }
15 // * * * * *
16
17 dimensions      [0 1 -1 0 0 0 0];
18
19 variable
20 {
21     high      (10 0 0);
22     middle    (5 0 0);
23     low       (1 0 0);
24 }
25
26 internalField    uniform (0 0 0);
27
28 boundaryField
29 {
30     movingWall
31     {
32         type      fixedValue;
33         value      uniform $:variable.high;
34     }
35     fixedWalls
36     {
37         type      fixedValue;
38         value      uniform (0 0 0);
39     }
40     frontAndBack
41     {
42         type      empty;
43     }
44 }
45
```



	define patch at constant/. (boundary)	U	
field type dimensions		volVectorField; [0 1 -1 0 0 0 0];	volScala [0 2 -2
internal Field		uniform (0 0 0);	uniform
movingWall	type wall; inGroups 2(wall wallM);	type fixedValue; value uniform (10 0 0);	type zer
fixedWalls	type wall; inGroups 1(wall);	type fixedValue; value uniform (0 0 0);	type zer
frontAndBa ck	type empty; inGroups 1(empty);	type empty;	type emp

MovingWallには、最終的に(10 0 0)が設定される。

```

31 variable
32 {
33     high      (10 0 0);
34     middle    (5 0 0);
35     low       (1 0 0);
36 }
37 a
38 {
39     a  $:variable.high;
40 }
41 aa
42 {
43     a  $:a.a;
44 }
45
46 aaa
47 {
48     a  $:aa.a;
49 }
50
51 internalField      uniform (0 0 0);
52
53 boundaryField
54 {
55     test            $:aaa.a;
56     movingWall
57     {
58         type        fixedValue;
59         value        uniform $test;
60     }
61     fixedWalls
62     {

```

変数定義では、ネスティングが可能になる。

実行すると、OpenFOAMはちゃんと判断する。  
gridEditor側もネスティングが可能。

gridEditor側は、10回までのネスティング  
リミットをかけている

	define patch at constant/. (boundary)	U
field type dimensions		volVectorField; [0 1 -1 0 0 0 0];
otherNames		highVelocity (10 0 0); highVelocityField uniform \$highVelocity;
internal Field		uniform (0 0 0);
otherNames (boundary)		test (10 0 0);
movingWall	type wall; inGroups 2(wall wallM);	type fixedValue; value uniform \$test;
fixedWalls	type wall; inGroups 1(wall);	type fixedValue; value uniform (0 0 0);
frontAndBack	type empty; inGroups 1(empty);	type empty;

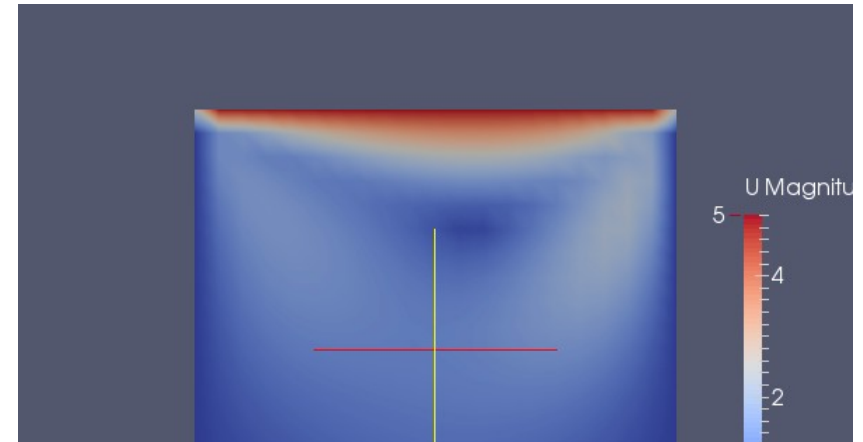


```

15 // *****
16
17 dimensions      [0 1 -1 0 0 0 0];
18
19 aaa
20 {
21     aa
22     {
23         b
24         {
25             c      (3 0 0);
26             d      (5 0 0);
27         }
28         a
29         {
30             type fixedValue;
31             value uniform (2 0 0);
32         }
33     }
34 }|
35
36 internalField      uniform (0 0 0);
37
38 boundaryField
39 {
40     movingWall
41     {
42         type fixedValue;
43         value uniform $:aaa.aa.b.d;
44     }
45
46     fixedWalls

```

階層型変数の段数は、2段以上でもOK。  
この例では、「\$:aaa.aa.b.d」で確認。



	define patch at constant/. (boundary)	U	
field type dimensions		volVectorField; [0 1 -1 0 0 0 0];	volSc [0 2
internal Field		uniform (0 0 0);	unifo
movingWall	type wall; inGroups 2(wall wallM);	type fixedValue; value uniform (5 0 0);	type
fixedWalls	type wall; inGroups 1(wall);	type fixedValue; value uniform (0 0 0);	type
frontAndBa ck	type empty; inGroups 1(empty);	type empty;	type

## まとめ

multiRegionについては、TreeFoamを使うことによって、領域分割後は、純粋なpatchだけに境界条件を設定すれば済む。後は、材料を設定すれば、計算する事ができる。

field内変数については、当初、記述の順番に変数を解釈すると思っていたが、変数解釈には、優先順位がある事がわかり、gridEditorを修正。  
(前回のバージョン2.3は、記述の順番で解釈している。)  
今回、「".\*"」や「\$:wallToRegion.U」を設定するに当たり、調べた結果、判明。  
今回のバージョン2.4より対応している。