

# 新TreeFoamの紹介

(TreeFoam ver 3.22.240225)

## <変更内容>

1. solver構成の違いに対応
2. snappyHexMeshの並列処理方法
3. FrontISTRとの連成解析
4. meshViewerのfaceZones表示
5. patchType 「mapped」 の設定内容が変更
6. controlDict内で「#includeIfPresent」の使用
7. regionPropertiesが無くなる
8. splitMeshRegionsのoptionが変更
9. multiRegionにおけるT fieldの境界条件が変更
10. multiRegion 「film」 への対応
11. pyFoamのインストール

# 1. solver構成の違いに対応

<従来(OF-10)>

solverは、controlDict内のapplicationで指示する。

```
/*-----* C++ *-----*\
=====
 \\      /  F i e l d
  \\    /   O p e r a t i o n
   \\  /    A n d
    \\ /     M a n i p u l a t i o n
 \\*-----*/
FoamFile
{
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// ***** //
```

```
application    icoFoam; solver
```

```
startFrom      startTime;
```

```
startTime      0;
```

```
stopAt         endTime;
```

```
endTime        0.5;
```

## <OF-11>

controlDict内のapplicationは、基本的に「foamRun」 or 「foamMultiRun」 しかない。  
項目として、「solver」 「regionSolvers」が追加。

```
/*-----* C++ *-----
=====
 \ /   F i e l d
  / \   O p e r a t i o n
 /   \   A n d
 \   /   M a n i p u l a t i o n
 \
 *
-----
FoamFile
{
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// *****

application    foamRun;
solver          incompressibleFluid; solver名
startFrom      startTime;
startTime      0;
```

```
/*-----* C++ *-----
=====
 \ /   F i e l d
  / \   O p e r a t i o n
 /   \   A n d
 \   /   M a n i p u l a t i o n
 \
 *
-----
FoamFile
{
    format      ascii;
    class       dictionary;
    object      controlDict;
}
// *****

application    foamMultiRun;
regionSolvers
{
    fluid       incompressibleFluid;
    solid       solid;
}

startFrom      startTime; region名とsolver名
startTime      0;
stopAt         endTime;
```



## 1-1. tutorials内のcase取得方法

「foamRun」「foamMultiRun」「その他」で選択していく。

tutorialsからcaseをコピーして、newCaseを作成します。  
caseを選択してください。

tutorials

場所を変更する

場所 /opt/openfoam11/tutorials 参照...

solver, caseを選択

区分	solver	case
foamRun:一般的なCFD	XiFluid	kivaTest
foamMultiRun:複数領域のCFD	compressibleMultiphaseVoF	moriyoshiHomogeneous
film	compressibleVoF	
legacy	fluid	
mesh	incompressibleDenseParticleFoam	
potentialFoam	incompressibleDriftFlux	
resources	incompressibleFluid	
	incompressibleMultiphaseVoF	
	incompressibleVoF	

<solverの内容>

キャンセル OK

foamRun →

foamMultiRun →

その他 →

## 1-2. TreeFoam内のsolver名表示方法

以下の考え方で表示する。

- foamRun : foamRunとsolverを表示
- foamMultiRun : solverが複数ある為、foamMultiRunのみを表示
- その他 : 「application」の内容を表示（従来どおりの表示）

### 表示例

Tree	solver	BCPn	nR	st	ed
> OF-10					
▼ OF-11					
> 01_cavity					
02_damBreak					
> 03_coolingSphere					
cavity_copy0					
> OF-9					
OF-v2206					
> myTutorials-bak					
> myTutorials_11					
> CAE-FrontISTR					
> CAE-OpenRadioss					

solver名表示	
<b>foamRun</b>	→ foamRun.incompressibleFluid anP 21 0.0 10.0
	→ foamRun.incompressibleVoF anP 21 0.0 1.0
<b>foamMultiRun</b>	→ foamMultiRun anP4 4 0.0 0.15
<b>その他</b>	→ icoFoam anP 6 0.0 0.5

## 2. snappyHexMeshの並列処理方法

TreeFoam上で、snappyHexMeshの並列処理ができなくなる。

<原因>

OF-11では、「reconstructParMesh」コマンドが無くなり、「reconstructPar」に統一。

このため、OF-11用の分岐を追加。

<従来(OF-10)>

以下を実行する。

```
blockMesh
decomposePar
mpirun -np 4 snappyHexmesh -parallel
reconstructParMesh -latestTime -cellDist
```

<OF-11>

最後をreconstructParに変更。

```
blockMesh
decomposePar
mpirun -np 4 snappyHexmesh -parallel
reconstructPar -latestTime
```

### 3. FrontISTRとの連成解析

連成解析する為のcodeは、OF-10と同じcodeが使えた。  
solver構成が異なっている為、tutorialsのcaseを再作成。

```
/*-----*- C++ -*-----
=====
\\      /  F ield
\\      /  O peration
\\      /  A nd
\\      /  M anipulation
\
*-----*-----
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       controlDict;
}
// *****

//application      pimpleFoam;
application        foamRun;

solver              incompressibleFluid;

startFrom           startTime;

startTime           0;

stopAt              endTime;
```

solverは、「pimpleFoam」でも動くが、  
新しいsolver構成に変更している。

solverが微妙に変わっているので、連成計算  
の定数（nStepsFsiなど）を変更して、  
caseを作成している。



## 4. meshViewerのfaceZones表示 meshViewerでfaceZoneが表示できなくなる。

<原因>

OF-11から、foamFileのascii書式が変わり、vtkOpenFOAMReader()が読めなくなる。  
meshViewer起動時にfaceZonesの内容をcheckし、「false,true」を「0,1」に修正するように変更。

<従来(OF-10)>

boolの値を「0」「1」で表現。

```
/*-----* C++ -*-----
      |
      |  F i e l d
      |  O p e r a t i o n
      |  A n d
      |  M a n i p u l a t i o n
      |
      |  OpenFOAM: The Open Source CFD
      |  Website: https://openfoam.or
      |  Version: 10
      |
      |-----*
\
*-----*
:
12373
12375
12381
12382
)
;
flipMap          List<bool>
512
(
1
1
1
:

```

<OF-11>

boolの値が「false」「true」で表現。

```
/*-----* C++ -*-----
      |
      |  F i e l d
      |  O p e r a t i o n
      |  A n d
      |  M a n i p u l a t i o n
      |
      |  OpenFOAM: The Open Source CFD
      |  Website: https://openfoam.or
      |  Version: 11
      |
      |-----*
\
*-----*
:
12373
12375
12381
12382
)
;
flipMap          List<bool>
512
(
true
true
true
:

```

## 5. patchType 「mapped」 の設定内容が変更

internalPatch (cyclic, mapped, baffle) の内、mappedでエラーが発生。

<原因>

OF-11でmappedの設定内容が変更されている。

OF-11用の分岐を追加。

<従来(OF-10)>

boundaryファイルの内容

```
8
(
  :
  inW_slave
  {
    type mappedPatch;
    inGroups List<word> 2(mappedPatch inW);
    sampleMode nearestPatchFace;
    sampleRegion region0;
    samplePatch inW_master;
    nFaces 80;
    startFace 62160;
  }
  :
)
```

<OF-11>

boundaryファイルの内容

```
8
(
  :
  inW_slave
  {
    type mapped;
    inGroups List<word> 2(mapped inW);
    neighbourRegion region0;
    neighbourPatch inW master;
    nFaces 80;
    startFace 62160;
  }
  :
)
```

## 6. controlDict内で「#includeIfPresent」の使用

controlDict内のapplication, startTimeなどの読込では、#includeを使っていなかった。

OF-11からregionSolversに#includeが使われる様になり、#includeを解釈するする様にした。

### controlDictの内容

```
/*----- C++ -----  
=====  
\\      /  F ield  
 \\    /   O peration  
  \\  /    A nd  
   \\ /    M anipulation  
  \\      /  
 \\    /  
  \\  /  
   \\ /  
=====  
\\  
*-----  
FoamFile  
{  
    format      ascii;  
    class       dictionary;  
    object      controlDict;  
}  
// * * * * *  
  
application    foamMultiRun;  
  
#includeIfPresent "regionSolvers"  
  
startFrom      startTime;  
startTime      0;  
stopAt         endTime;  
:
```

```
/*----- C++ -----  
=====  
\\      /  F ield  
 \\    /   O peration  
  \\  /    A nd  
   \\ /    M anipulation  
  \\      /  
 \\    /  
  \\  /  
   \\ /  
=====  
\\  
*-----  
FoamFile  
{  
    format      ascii;  
    class       dictionary;  
    object      controlDict;  
}  
// * * * * *  
  
application    foamMultiRun;  
  
//#includeIfPresent "regionSolvers"  
regionSolvers  
{  
    fluid      fluid;  
    solid      solid;  
}  
  
startFrom      start  
startTime      0;  
:
```

解釈した結果に書き換える。

## 7. regionPropertiesが無くなる

従来は、regionPropertiesを使って、multiRegionの各regionを流体or固体に分けていた。  
 OF-11では、controlDict内のregionSolversを確認して、流体or固体に分けることになる。

### <従来(OF-10)> regionProperties

```

/*----- C++ -----
=====
 \\  /   F ield          |   OpenFOAM: The Open Source CFD
  \\ /    O peration     |   Website:  https://openfoam.or
   \\ /      A nd        |   Version:  10
    \\ /        M anipulation
 \
 *-----
FoamFile
{
    format      ascii;
    class       dictionary;
    location    "constant";
    object      regionProperties;
}
// *****

regions
2
(
    region名
    fluid 1 ( fluid )
    solid 1 ( solid )
);
  
```

### <OF-11> controlDict

```

/*----- C++ -----
=====
 \\  /   F ield          |   OpenFOAM: The Open Source CFD
  \\ /    O peration     |   Website:  https://openfoam.or
   \\ /      A nd        |   Version:  11
    \\ /        M anipulation
 \
 *-----
FoamFile
{
    format      ascii;
    class       dictionary;
    object      controlDict;
}
// *****

application      foamMultiRun;

regionSolvers
{
    region名      solver名
    fluid          incompressibleFluid;
    solid          solid;
}
  
```

solver名で流体or固体を判断

## 8. splitMeshRegionsのoptionが変更

作成したcellZoneを使って、regionを作成する時、「splitMeshRegions」コマンドを使っているがこのoptionがOF-11から変更されており、エラーが発生する。

<従来(OF-10)>

以下を実行する。

```
splitMeshRegions -cellZonesOnly
```

「-cellZonesOnly」のoptionが無くなった。

<OF-11>

以下を実行する。

```
splitMeshRegions -cellZones
```

「-cellZonesOnly」の代わりに  
「-cellZones」に使う。

## 9. multiRegionにおけるT fieldの境界条件

OF-11から流体・固体のT field同士で値を交換するpatchの境界条件が変更されている。

<従来(OF-10)>

流体側のpatch

```
type compressible::turbulentTemperatureCoupledBaffleMixed;  
neighbourFieldName T;  
kappa kappa;  
Tnbr T;  
value uniform $iniTemp;
```

固体側のpatch

```
type compressible::turbulentTemperatureCoupledBaffleMixed;  
value uniform $iniTemp;  
Tnbr T;  
kappa kappa;
```

multiRegion用のsolverは「chtMultiRegionFoam」のみであり、流体固体の熱連成計算は、「chtMultiRegionFoam」を使う。

<OF-11>

流体側のpatch

```
type coupledTemperature;  
value uniform $iniTemp;
```

固体側のpatch

```
type coupledTemperature;  
value uniform $iniTemp;
```

OF-11の場合、regionSolversで流体solverと固体solverを指定する為、solverの組み合わせは、多数存在する。この為、複雑なmultiRegionの計算が可能になっている。

## 10. multiRegion 「film」 への対応

OF-11から「film」が追加されている。

従来の流体・固体のmultiRegionの場合は、case直下にdefaultRegion (region0) が存在するが、filmの場合は、case直下には、defaultRegionが存在しない。(case直下にconstant/polyMeshが無い)

### <通常のmultiRegion>

```
caseName/  
  0/  
    fluid/  
    solid/  
    constant/  
    fluid/  
    solid/  
    polyMesh/ ← polyMeshが存在する。  
    system/  
    fluid/  
    solid/
```

Case  
 defaultRegion  
 Fluid region  
 Solid region

### <filmのmultiRegion>

```
caseName/  
  0/  
    fluid/  
    solid/  
    constant/  
    fluid/  
    solid/      polyMeshが無い  
    system/  
    fluid/  
    solid/
```

Case  
 Fluid region  
 Solid region

TreeFoam上でmultiRegionを扱う場合、defaultRegionが存在する事を前提にしている。

この為、filmのmultiRegionを解析caseに設定し、multiRegion関係の操作する、meshViewerでpatch形状を確認する時、エラーが発生。

defaultRegionが存在しないmultiRegionがある事を考慮する様に修正。

## 11. pyFoamのインストール

pyFoamのplotWatcherは、OpenFOAMが出力するlog内容を読み取り、残渣をグラフ表示する。

OpenFOAMがversionUpにより、logのフォーマットが変わっており、

旧のpyFoamのままでは、残渣のグラフ表示時に警告やエラーが発生する。

### <foamRunのlog>

```
Starting time loop
Time = 0.005s
Courant Number mean: 0 max: 0
smoothSolver: Solving for Ux, Initial residual = 1, Final
smoothSolver: Solving for Uy, Initial residual = 0, Final
DICPCG: Solving for p, Initial residual = 1, Final residua
time step continuity errors : sum local = 0.000466513, glob
DICPCG: Solving for p, Initial residual = 0.590864, Final
time step continuity errors : sum local = 2.74685e-09, glob
ExecutionTime = 0.006418 s ClockTime = 0 s
Time = 0.01s
:
```

“s”が追加 (OF-10から)  
エラー発生し、グラフ表示しない

“s”が追加 (OF-6から)  
毎回警告が発生する

### <foamMultiRunのlog>

```
Starting time loop
fluid Courant Number mean: 0 max: 0
solid Diffusion Number mean: 8.1347725 max: 955.02803
Time = 0.0005s
fluid diagonal: Solving for rho, Initial residual = 0, Fin
fluid DILUPBiCGStab: Solving for Ux, Initial residual = 1,
fluid DILUPBiCGStab: Solving for Uy, Initial residual = 1,
fluid DILUPBiCGStab: Solving for Uz, Initial residual = 1,
fluid DILUPBiCGStab: Solving for h, Initial residual = 1,
solid DICPCG: Solving for e, Initial residual = 1, Final r
fluid GAMG: Solving for p_rgh, Initial residual = 0.996276
fluid GAMG: Solving for p_rgh, Initial residual = 0.042365
:
```

logにregion名が追加 (OF-11から)

TreeFoam側でlogを成形 (“s”を削除) して  
logファイルを作成して対応していた。

実行例:

```
icoFoam | teeCorrect.py solve.log
```

「teeCorreocr.py」が成形して、solve.logにlogを保存

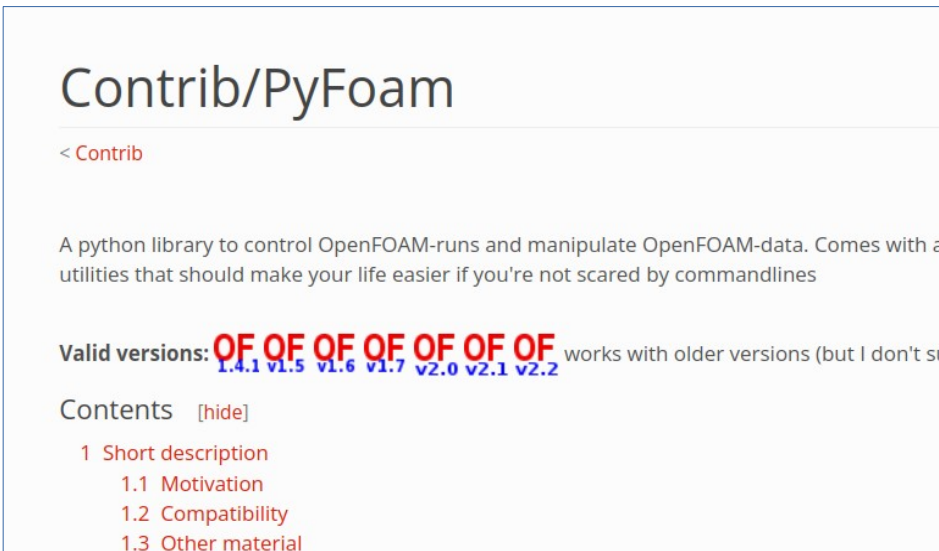
新たに対応をする必要がある。



従来からのpyFoamは、「https://openfoamwiki.net/index.php/Contrib/PyFoam」から入手していたが、数年前からpyFoamのversionUpが止まっていた。

しかし、新しいweb「https://pypi.org/project/PyFoam/」から最新版が入手できる事が判明。

### <旧のweb画面>



Contrib/PyFoam

< Contrib

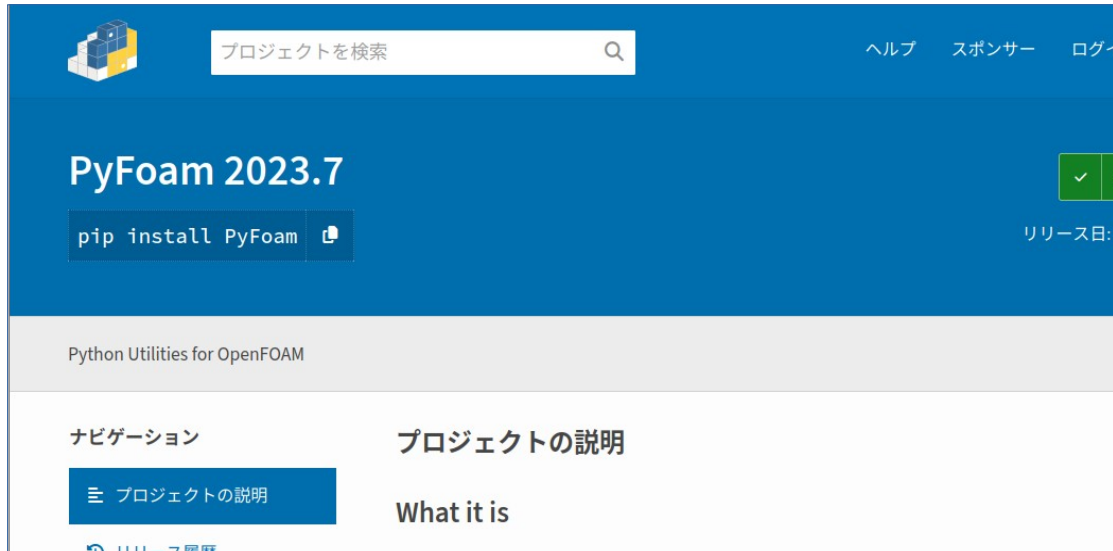
A python library to control OpenFOAM-runs and manipulate OpenFOAM-data. Comes with a utilities that should make your life easier if you're not scared by commandlines

Valid versions: **OF OF OF OF OF OF** works with older versions (but I don't su  
1.4.1 v1.5 v1.6 v1.7 v2.0 v2.1 v2.2

Contents [hide]

- 1 Short description
  - 1.1 Motivation
  - 1.2 Compatibility
  - 1.3 Other material

### <新しいweb画面>



PyFoam 2023.7

pip install PyFoam

Python Utilities for OpenFOAM

ナビゲーション

- プロジェクトの説明

プロジェクトの説明

What it is

リリース履歴

最新版のpyFoamを「pip install PyFoam」でインストールした結果、OF-11のlog内容を読み取り、正常にグラフ表示できた。

これにより、TreeFoam側で、logの成形を廃止している。

icoFoam | tee solve.log