

swak4Foam(funkySetFields, groovyBC)を 使ってみました(2)

～関数による境界条件の設定～

勉強会@関西幹事
富原 大介

FunkySetBoundaryFields

Examples/FunkySetBoundaryFields/ 内にある例題
funkySetBoundaryDict.dambreak を解読する

基本事項

FunkySetBoundaryFieldsはFunkySetFieldsと同じく、
コマンドラインからでもDictを用いても実行することができる。

FunkySetFieldsは場（internalField）の値をセットするのに対し、
FunkySetBoundaryFieldsは境界条件（patch）の値をセットすることができる。

→同じく境界条件をセットするgroovyBCとの違いは、
実行時ライブラリを参照するかどうか

実行時

systemにfunkySetBoundaryDictを設置

```
$ funkySetBoundaryFields -time 0
```

※[-time 0]は省略可

FunkySetBoundaryFields

ヘッダ

```

/*-----*- C++ -*-----*\
| ===== |
| \\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox
| \\      /  O peration  | Version: 1.7.1
|  \\    /   A nd        | Web:      www.OpenFOAM.com
|  \\    /   M anipulation |
\*-----*-*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       controlDict;
}
// ***** //
    
```

objectがcontrolDictになっているが、
funkySetBoundaryDictとしても特に問題ない
□ゴもなくてOK

FunkySetBoundaryFields

境界条件 (1)

```
velocities {
  field U;
  expressions
  (
    {
      target value;
      patchName leftWall;
      variables "maxY=max(pts().y);thres=0.25*maxY;";
      expression "(pos().y<thres) ? vector(1,0,0)*(maxY-pos().y) : vector(0,0,0)";
    }
    {
      target value;
      patchName rightWall;
      variables "maxY=max(pts().y);";
      expression "vector(0,1,0)*(maxY-pos().y+p_rgh)";
    }
  );
}
```

velocities : タイトル (任意)。セットする境界条件のファイルごとにタイトルをつける。

field U : 「field」はここではファイル名。「field /U」で「0/U」ファイルを指す。

expressions : この中にセットする条件を記述する。

FunkySetBoundaryFields

境界条件 (1)

```
target value;
patchName leftWall;
variables "maxY=max(pts().y);thres=0.25*maxY;";
expression "(pos().y<thres) ? vector(1,0,0)*(maxY-pos().y) : vector(0,0,0)";
```

target value : 境界条件の中の「value」の値をセットするということ。

patchName leftWall : patch「leftWall」を指定。

variables : expressionで使用する変数を記述する。

※ "(式)"の中に[;]が入っていることに注意

expression : 条件式を記述する。

元の0/Uファイル

```
boundaryField
{
    leftWall
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }
}
```

ここが変更される

max() : 最大値

pts().yおよびpos().y : y座標

※pts()とpos()の違い

pts()は頂点での値でpos()は「Vector field with the cell-centers」(要素中心)の値。

Uは要素中心の値を設定するようになっている。よって「expression "vector(1,0,0)*(pts().y)"」はNG。

A?B:C : IF文。AがtrueならばB、falseならばC
vector(1,0,0)*(maxY-pos().y)というように、ベクトル×スカラーの計算もOK

FunkySetBoundaryFields

境界条件 (2)

複数のファイルにまたがって変更する場合は、このように分けて書く。

```
velocities {
  field U;
  .....
}
dropIn {
  field alpha1;
  expressions
  (
    {
      target inletValue;
      patchName atmosphere;
      variables "maxX=max(pts().x);thres1=0.4*maxX;thres2=0.6*maxX;";
      expression "(pos().x>thres1 && pos().x<thres2) ? 1 : 0";
    }
  );
}
```

元の0/alpha1ファイル

```
atmosphere
{
  type inletOutlet;
  inletValue uniform (0 0 0);
  value uniform (0 0 0);
}
```

ここが変更される

A && B : AかつB

FunkySetFields

Examples/FunkySetFields/ 内にある例題

setLowerHalfAndATube.funkySetFieldsDict を解読する

基本事項

FunkySetFieldsは、
コマンドラインからでもDictを用いても実行することができる。

基本的にFunkySetFieldsは場（internalField）の値をセットするものであるが、
境界条件（patch）の値をセットすることもできる。
→ fixedValueのみ

実行時

system()にfunkySetDictを設置

```
$ funkySetFields -time 0
```

※[-time 0]は**省略不可**

FunkySetFields

ヘッダ

```
FoamFile
{
  version      2.0;
  format       ascii;
  root         "/.automount/werner/Werner/bgschaid/bgschaid-foamStuff/DepositionTests";
  case         "neutralTest";
  instance     "system";
  local        "";
  class        dictionary;
  object       funkySetFieldsDict;
}
```

おそらくあまり決まりはない

FunkySetFields

概要

```
expressions
(
  initAlpha1
  {
    field alpha1;
    .....
  }
  clearAlpha1
  {
    field alpha1;
    .....
  }
  setTube
  {
    field tubeField;
    .....
  }
);
```

引き続きdamBreakを使用することを想定しています。
（解説のため境界名等一部変更）

変更するファイルごとに作成する

expressions : この中にセットする条件を記述する。
initAlpha1, clearAlpha1, setTube : タイトル（任意）。
field alpha1 : 「field」はファイル名。

FunkySetFields

境界条件の変更

```
initAlpha1
{
    field alpha1;
    expression "1";
    condition "pos().y<0.3";
    valuePatches ( leftWall );
    keepPatches true;
}
clearAlpha1
{
    field alpha1;
    expression "0";
    keepPatches true;
}
```

expression : セットする値。
condition : 条件式。
valuePatches : 境界条件 (patch) 名
keepPatches : 境界条件を保持する (後述)

※「variables」を使用して変数を設定することも可能

FunkySetFieldsでも境界条件を設定することができる。
(ただし、fixedValueのみ。)

FunkySetFields

境界条件の変更

手順

① 「valuePatches」を用いて境界条件(patch)の値をセットする（intiAlpha1）

→このとき、一緒に「internalField」の値も変更される

② 「internalField」の値を元の値にセットする（clearAlpha1）

→その際、「keepPatches true;」を用いて①でセットした境界条件の値を保持する

FunkySetFields

ファイルの新規作成

```
setTube
{
    field tubeField;
    create true;
    expression "pos().z";
    condition "sqrt(pow(pos().x,2)+pow(pos().y,2))<0.0001";
    dimension [1 2 3 4 5 6 7];
}
```

「create true;」：新しくファイルを作成する場合に記述する。
※この場合「0/tubeField」ファイルが作成される。これがないとエラーとなる。

dimension：次元。たとえば「m/s」なら[0 1 -1 0 0 0 0]。

「field」で存在しないファイルを指定した場合、
新しくファイルが作成される。

FunkySetFields

FunkySetFieldsを使用する際の境界条件に関する注意

FunkySetFieldsを使用すると「keepPatches true」としない限り、
境界条件が「zeroGradient」にすべて変更される。
（対称条件などの一部のpatchを除く）

ファイルを新規に作成する際も境界条件はすべて「zeroGradient」。

また、FunkySetBoundaryFieldsと異なり、
境界条件はfixedValueのvalueのみ変更可能（前述）。
その他（例えば「inletValue」のvalueなど）を指定すると、
境界条件はfixedValueに上書きされる。