

## 目次

1	m4 を用いた blockMesh の作成 .....	2
1.1	m4 とは .....	2
1.2	m4 の使用方法 .....	2
1.3	blockMesh への m4 の適用メリット .....	2
1.4	m4 の適用例：円管流れ解析用の blockMesh の作成 .....	3
1.4.1	m4 ファイルの内容 .....	3
1.4.2	blockMeshDict ファイルの内容 .....	5
1.5	参考にしたページ .....	7
2	swak4Foam のインストール .....	8
2.1	swak4Foam とは .....	8
2.2	swak4Foam のインストール手順 .....	8
2.2.1	事前準備 .....	8
2.2.2	Swak4Foam の取得とコンパイル .....	8
2.3	groovyBC (swak4Foam) の使用例 .....	8
2.3.1	事前準備 .....	8
2.3.2	使用例 .....	9
2.3.3	解析結果 .....	11
2.4	参考にしたページ .....	12

## 1 m4 を用いた blockMesh の作成

### 1.1 m4 とは

UNIX の標準コマンドで OpenFOAM®用ではない。文字列を操作する場合に使うコマンドのようだ。

### 1.2 m4 の使用方法

以下ディレクトリにサンプルファイルがある。

```
$FOAM_RUN/tutorials/incompressible/MRFSimpleFoam/mixer/vessel2D
```

実際の使用には、以下のコマンドをターミナルに入力する。m4 に sample.m4 というファイルを入力し、blockMeshDict ファイルに出力するという意味。(図 1)

```
$ m4 < constant/polyMesh/sample.m4 > constant/polyMesh/blockMeshDict
```

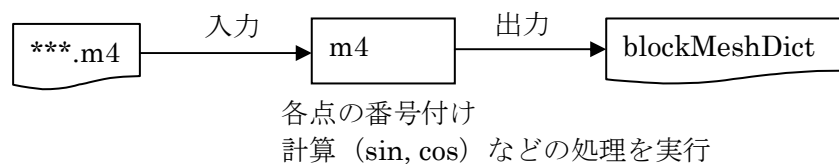


図 1. m4 の動作の概略図

### 1.3 blockMesh への m4 の適用メリット

OpenFOAM®では blockMesh 作成時に各点に番号付け (ラベリング) されるが、m4 を用いると自動的にラベリングをしてくれるので便利である。例えば、一度 blockMeshDict を作成した後に点を 1 つ追加すると、各点の番号も変更されてしまい、面倒な作業になる場合がある。m4 を用いれば、自動で番号付けをしてくれるので、この作業がなくなるので、便利である。

## 1.4 m4 の適用例：円管流れ解析用の blockMesh の作成

m4 のテストとして、図 2 に示す円管のメッシュを作成する。

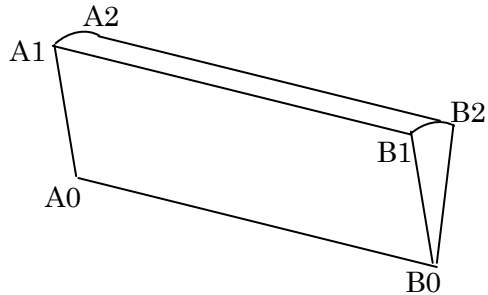


図 2 円管 (5° にスライスしたもの) の模式図

### 条件

円管長さ L : 200mm  
円管直径 D : 10mm、円管半径 : 5mm  
軸対称計算 (wedge patch を使用)  
A0~A2, B0~B2 は各頂点を表す

### 1.4.1 m4 ファイルの内容

以下に m4 ファイルを記す。

```
/*-----* C++ *-----*¥
|=====|
| ¥¥ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| ¥¥ / O p e r a t i o n | Version: 1.7.1 |
| ¥¥ / A n d | Web: www.OpenFOAM.com |
| ¥¥/ M a n i p u l a t i o n | |
¥*-----*/
FoamFile
{
    version 2.0;
    `format' ascii;
    class dictionary;
    object blockMeshDict;
}
// ***** //
// General macros to create 2D/extruded-2D meshes
// Usage: m4 < constant/polyMesh/blockMeshDict.m4 > constant/polyMesh/blockMeshDict

changequote(//)changequote(,)
define(calc, [systemcmd(perl -e 'print ($1)'])
define(VCOUNT, 0)
define(vlabel, [[// ]Vertex $1 = VCOUNT define($1, VCOUNT)define([VCOUNT],
incr(VCOUNT))])
```

ここまでは tutorial のファイルそのまま  
vlabel が VCOUNT に置き換えられて、インクリメントしていくようだ。

```

define(pi, 3.14159265) //sin, cos の計算に使用するため、pi を定義する。

// ****

// Length of Pipe //円管長さの定義
define(L, 200) //20D

// Inner diameter/radius of Pipe //円管直径と半径の定義
define(d0, 10)
define(r0, calc(d0/2))

// angle of theta
define(theta, 2.5)

// Number of cells between planeA and planeB (axial) //円管長手方向の分割数
define(Ldx, 200) //20D

// Number of cells between plane and Plane (radial) //円管半径方向の分割数
define(rdy, 20)

// ****

convertToMeters 0.001;

define(xA, -L) //xA (A の X 座標位置として、-L が定義される)
define(yA1, calc(r0*cos((pi/180)*theta))) //A1 の Y 座標が定義される。
//yA2=yA1

define(zA1, calc(r0*sin((pi/180)*theta))) //A1 の Z 座標が定義される。
//zA2=-zA1

define(xB, 0) //B の X 座標(=0)を定義する。
//yB1=yB2=yA1 //yB1, yB2 (B1 と B2 の Y 座標) は yA1 と同一なので、定義しない。
//zB1=zB2=-zA1

// ****

vertices
(
  (xA 0 0) vlabel(A0) // A0 の座標の定義 この場合は(-200 0 0)となる
  (xA yA1 -zA1) vlabel(A1)
  (xA yA1 zA1) vlabel(A2)

  (xB 0 0) vlabel(B0)
  (xB yA1 -zA1) vlabel(B1)
  (xB yA1 zA1) vlabel(B2)
);

blocks
(
  //block0 ブロックの定義 上記で使用した A0~A2, B0~B2 をそのまま使用して良い。
  hex (B0 A0 A2 B2 B0 A0 A1 B1)(Ldx rdy 1)simpleGrading (1 0.2 1)
);

edges

```

```

(
);

patches
(
    patch inlet
    (
        (A0 A2 A1 A0)
    )

    patch outlet
    (
        (B0 B2 B1 B0)
    )

    wall nozzle_wall
    (
        (A1 A2 B2 B1)
    )

    wedge back_face
    (
        (A0 A1 B1 B0)
    )

    wedge front_face
    (
        (A0 A2 B2 B0)
    )

    symmetryPlane axis
    (
        (B0 A0 A0 B0)
    )

);

```

#### 1.4.2 blockMeshDict ファイルの内容

以下に、上記 m4 ファイルから作成された `blockMeshDict` ファイルの中身を記す。「//」でコメントアウトしていた語句はそのまま残る。各点の座標が計算され数値に置き換わっていることが分かる。また `vlabel` で付けた名称に番号付けがされていることが分かる。たとえば `A0` が `Vertex A0 = 0` となっている。

```

// *****

/*----- C++ -----*/
|=====|
| ¥¥ / Field | OpenFOAM: The Open Source CFD Toolbox |
| ¥¥ / Operation | Version: 1.7.1 |
| ¥¥ / And | Web: www.OpenFOAM.com |
| ¥¥/ Manipulation |
¥*-----*/

```

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}
// ***** //
// General macros to create 2D/extruded-2D meshes
// Usage: m4 < constant/polyMesh/blockMeshDict.m4 > constant/polyMesh/blockMeshDict

// ***** //

// Length of Pipe

// Inner diameter/radius of Pipe

// angle of theta

// Number of cells between plane and plane (axial)

// Number of cells between plane and Plane (radial)
// ***** //
convertToMeters 0.001;

//yA2=yA1
//zA2=-zA1

//yB1=yB2=yA1
//zB1=zB2=-zA1

// ***** //

vertices
(
    (-200 0 0) // Vertex A0 = 0
    (-200 4.99524110792016 -0.218096936577626) // Vertex A1 = 1
    (-200 4.99524110792016 0.218096936577626) // Vertex A2 = 2

    (0 0 0) // Vertex B0 = 3
    (0 4.99524110792016 -0.218096936577626) // Vertex B1 = 4
    (0 4.99524110792016 0.218096936577626) // Vertex B2 = 5
);

blocks
(
    //block0
    hex (3 0 2 5 3 0 1 4)(200 20 1)simpleGrading (1 0.2 1)
);

edges
(
);

patches
(
    patch inlet

```

```

    (
      (0 2 1 0)
    )

    patch outlet
    (
      (3 5 4 3)
    )

    wall nozzle_wall
  (
    (1 2 5 4)
  )

    wedge back_face
    (
      (0 1 4 3)
    )

    wedge front_face
    (
      (0 2 5 3)
    )

    symmetryPlane axis
    (
      (3 0 0 3)
    )

);

// ***** //

```

## 1.5 参考にしたページ

- Mogurana's Blog  
<http://mogura7.zenno.info/~et/wordpress/2011/07/08/blockmesh/>
- Chalmers univ. Open source CDF course  
[http://www.tfd.chalmers.se/~hani/kurser/OS\\_CFD\\_2010/erwinAdiHartono/erwinAdiHartonoReport.pdf](http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2010/erwinAdiHartono/erwinAdiHartonoReport.pdf)

## 2 swak4Foam のインストール

### 2.1 swak4Foam とは

funkySetFields と groovyBC が合わさったライブラリである。様々なポスト処理へ使用できる simpleFunction 等のライブラリも含まれている。

### 2.2 swak4Foam のインストール手順

#### 2.2.1 事前準備

まず、以下のパッケージが必要である。

- ファイルを checkout するための subversion
- コンパイルのための bison, flex (flex は DEXCS2010 にインストール済みだった)

上記パッケージをインストールする。

```
$ sudo apt-get install subversion
```

```
$ sudo apt-get install bison
```

#### 2.2.2 Swak4Foam の取得とコンパイル

このあと、以下コマンドを順番に実行する。

```
$ run
```

```
$ cd ..
```

```
$ mkdir Libraries
```

```
$ cd Libraries
```

```
$
```

```
svn
```

```
checkout
```

```
https://openfoam-extend.svn.sourceforge.net/svnroot/openfoam-extend/trunk/Breeder_1.7/libraries/swak4Foam/
```

```
$ cd swak4Foam
```

```
$ wmake all
```

※) 2011/10/3 のアップデートにより、simpleFunctionObject も含まれるようになった。以前は別途取得する必要があったが、最新版では取得の必要はなくなった。

### 2.3 groovyBC (swak4Foam) の使用例

#### 2.3.1 事前準備

controlDict ファイル内の一番下に以下のライブラリへのリンクを追加する。

```
libs ("libOpenFOAM.so" "libgroovyBC.so"); //Add groovyBC lib
```



### 2.3.2 使用例

groovyBC のテストとして、円管内乱流の解析を行ってみました。通常十分発達した流れとなるには、通常、助走区間が 20D 程度（直径の 20 倍）必要とされている。

そこで、カルマン-プラントルの 1/7 乗則（1 式）を境界条件として設定し、助走区間が短くても発達した流れとなるか試してみました。

$$u = u_{\max} \times \left(1 - \frac{r}{r_0}\right) \cdot \cdot \cdot (1)$$

$r_0$  : 円管半径

$r$  : 壁面からの距離

$u_{\max}$  : 最大速度

表 1 のケースにて、groovyBC が機能しているかまた、1/7 乗則に効果があるかを確認した。

表 1 解析条件

	Case01	Case02	Case03
直径 D	10mm	10mm	10mm
長さ L	50mm (5 D)	50mm (5 D)	100mm (20 D)
軸方向分割数 (X 方向)	50	50	200
半径方向分割数 (Y 方向)	20	20	20
U の境界条件	uniform	groovyBC (1/7 乗則)	uniform

RANS : simpleFoam (standard k-epsilon)

divScheme : linear, relaxationFactors (p, U, k, epsilon) = (0.3, 0.5, 0.4, 0.4)

具体的には、0 ディレクトリの U ファイルへ以下のように記述すれば使用することができる。  
U ファイルの内容を下記に記す。

```

/*-----* C++ *-----*
| ===== |
| ¥¥ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| ¥¥ / O p e r a t i o n | Version: 1.7.1 |
| ¥¥ / A n d | Web: www.OpenFOAM.com |
| ¥¥/ M a n i p u l a t i o n | |
¥*-----*/

```

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       U;
}
// ***** //

dimensions      [0 1 -1 0 0 0];
internalField   uniform (0 0 0);

boundaryField
{
    inlet
    {
        type          groovyBC; //groovyBC を使用する。
        value          uniform (0 0 0); // 最初のステップの値 (これがないとエラーが出
る)
        valueExpression "vector(26*pow((1-pos().y*200),0.142857142857),0,0)"; //26 ×
pow(1-r/r0), 1/7), 0, 0)
    }

    outlet
    {
        type          zeroGradient;
    }
    inletOutlet
    {
        type          zeroGradient;
    }

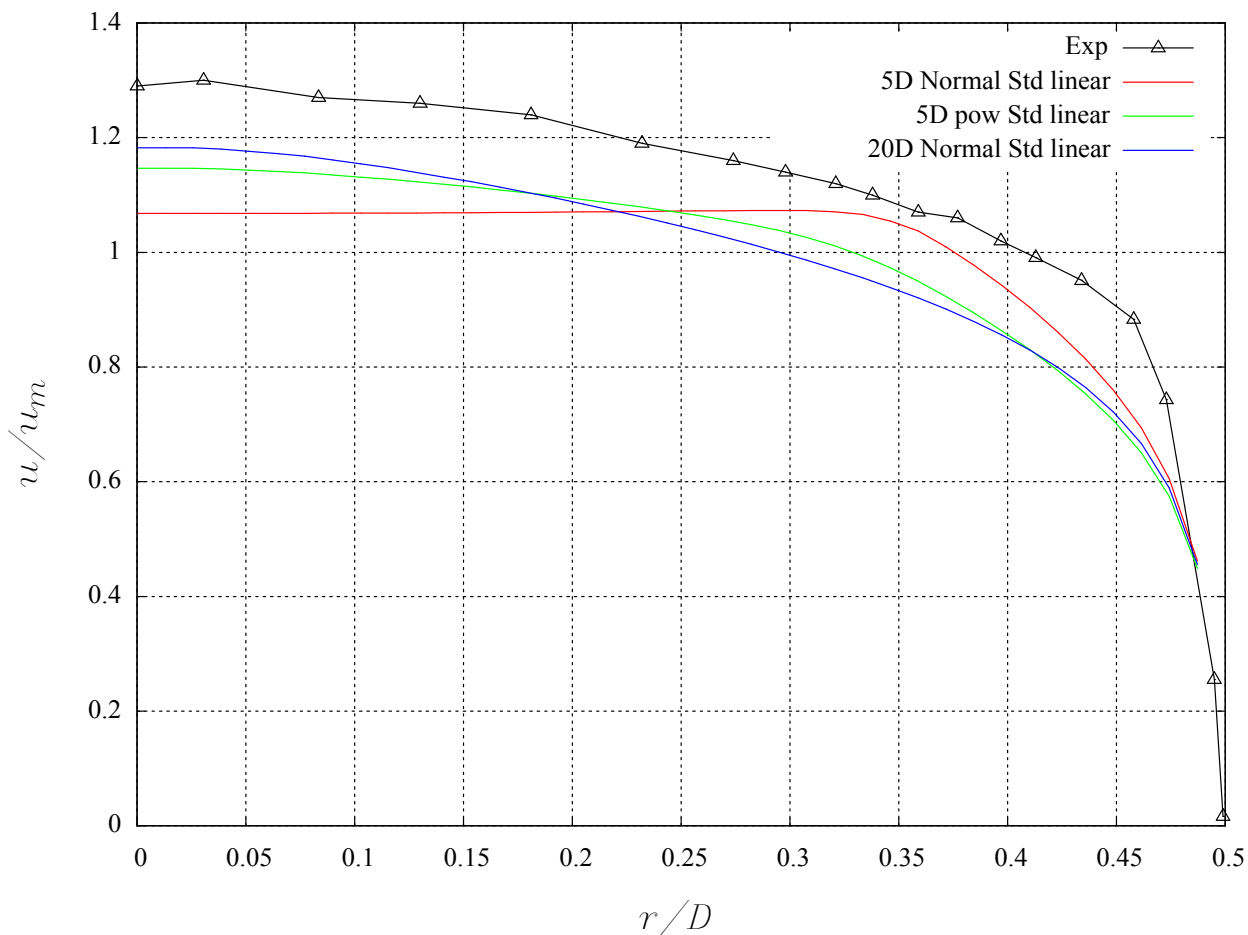
    nozzle_wall
    {
        type          fixedValue;
        value          uniform (0 0 0);
    }

    back_face
    {
        type          wedge;
    }
    front_face
    {
        type          wedge;
    }
    axis
    {
        type          symmetryPlane;
    }
}

// ***** //

```

### 2.3.3 解析結果



グラフ 1 出口での流速分布

実験結果 (Exp) は、「オリフィス自由噴流の流動解析機論 (B) 74 巻 737 号(2008)」より。ただし、 $0.2 D$  での結果のため、あくまで参考である。

- Case01 の結果 (赤) より Case02 (黄緑) の方が、Case03 の結果 (青) に近づいている。結果から境界条件に  $1/7$  乗則を当てはめることで、解析領域を小さくできるといった一定の効果はあると予想される。ただし、最大速度や境界層付近の速度分布は一致しなかった。

## 2.4 参考にしたページ

- OpenFOAM Wiki

<http://openfoamwiki.net/index.php/Contrib/swak4Foam>

さらにこのページの 5.1 Further information にある 2 つの PDF ファイル

[http://www.openfoamworkshop.org/6th\\_OpenFOAM\\_Workshop\\_2011/Program/Training/gschaider\\_slides.pdf](http://www.openfoamworkshop.org/6th_OpenFOAM_Workshop_2011/Program/Training/gschaider_slides.pdf)

<http://openfoamwiki.net/index.php/File:Swak4FoamLeoben2011.pdf>

- OpenFOAM 勉強会 for beginner (関西)富原さんの発表資料

<http://goo.gl/q3EHu>