OpenFOAM勉強会 for beginner 2期 第1回

OpenFOAMの使い方

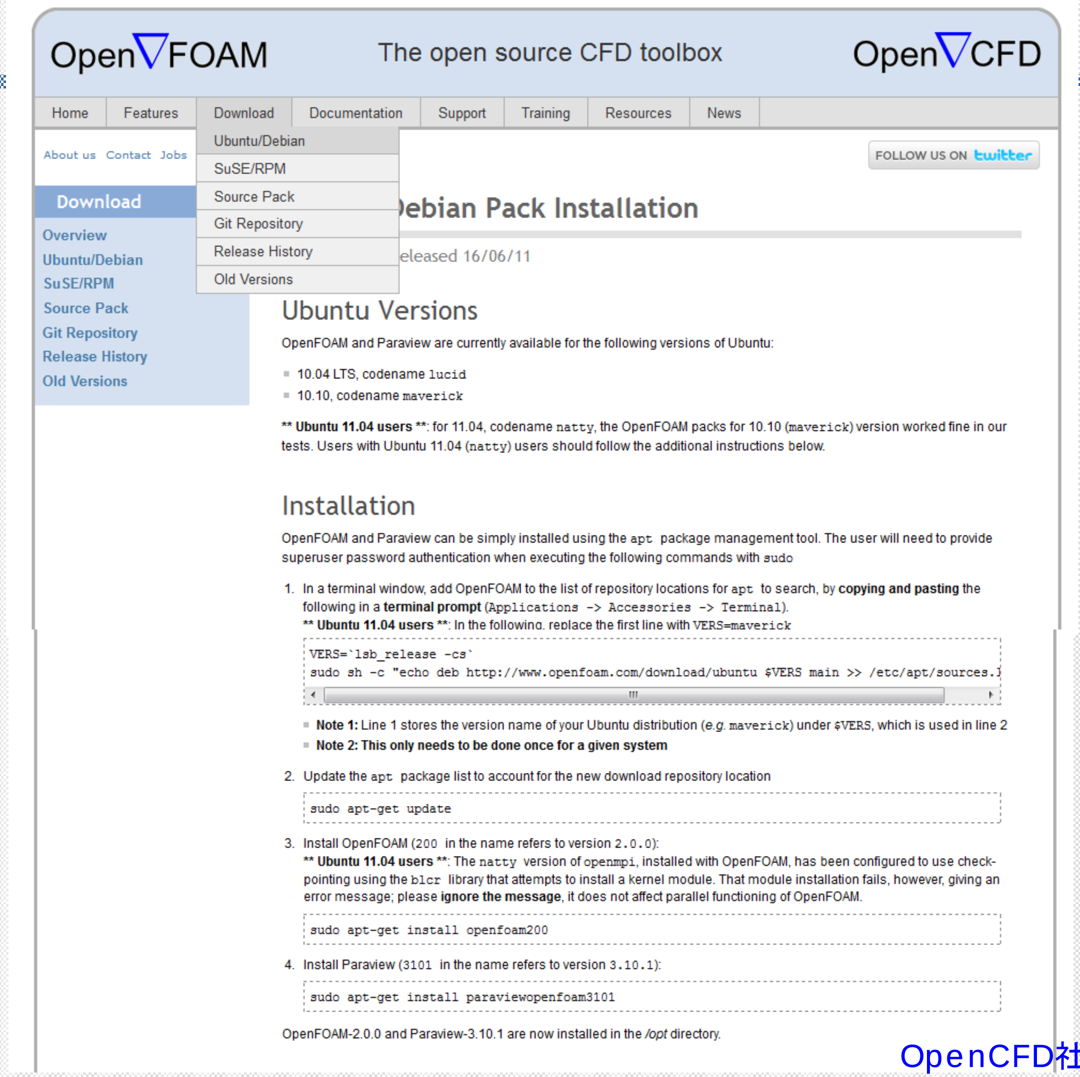
柴田　貴裕

目標

- 既にOpenFOAMはインストール済み

- さまざまなチュートリアルに取り組めるように チュートリアルの実行の方法を scalarTransportFoamを例に用いて見ていく

(インストールはOpenCFDのHPの方法に従えば比較的容易にできる)


Open∇FOAM The open source CFD toolbox Open∇CFD

Home | Features | Download | Documentation | Support | Training | Resources | News

Ubuntu/Debian
SuSE/RPM
Source Pack
Git Repository
Release History
Old Versions

About us Contact Jobs

FOLLOW US ON twitter

Download
Overview
Ubuntu/Debian
SuSE/RPM
Source Pack
Git Repository
Release History
Old Versions

ebian Pack Installation

eleased 16/06/11

Ubuntu Versions

OpenFOAM and Paraview are currently available for the following versions of Ubuntu:

- 10.04 LTS, codename lucid
- 10.10, codename maverick

** **Ubuntu 11.04 users** **: for 11.04, codename natty, the OpenFOAM packs for 10.10 (maverick) version worked fine in our tests. Users with Ubuntu 11.04 (natty) users should follow the additional instructions below.

Installation

OpenFOAM and Paraview can be simply installed using the apt package management tool. The user will need to provide superuser password authentication when executing the following commands with sudo

1. In a terminal window, add OpenFOAM to the list of repository locations for apt to search, by **copying and pasting** the following in a **terminal prompt** (Applications -> Accessories -> Terminal).
** **Ubuntu 11.04 users** **: In the following, replace the first line with VERS=maverick

```
VERS=`lsb_release -cs`
sudo sh -c "echo deb http://www.openfoam.com/download/ubuntu $VERS main >> /etc/apt/sources.l
```

- **Note 1:** Line 1 stores the version name of your Ubuntu distribution (*e.g.* maverick) under $VERS, which is used in line 2
- **Note 2: This only needs to be done once for a given system**

2. Update the apt package list to account for the new download repository location

```
sudo apt-get update
```

3. Install OpenFOAM (200 in the name refers to version 2.0.0):
** **Ubuntu 11.04 users** **: The natty version of openmpi, installed with OpenFOAM, has been configured to use check-pointing using the blcr library that attempts to install a kernel module. That module installation fails, however, giving an error message; please **ignore the message**, it does not affect parallel functioning of OpenFOAM.

```
sudo apt-get install openfoam200
```

4. Install Paraview (3101 in the name refers to version 3.10.1):

```
sudo apt-get install paraviewopenfoam3101
```

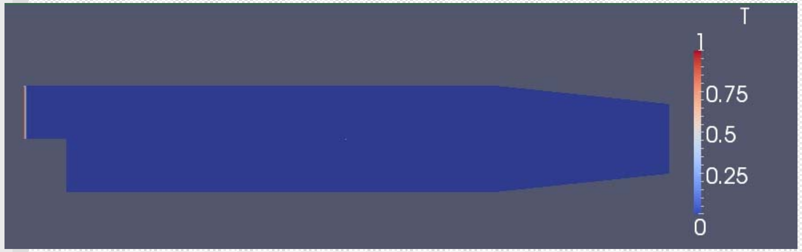
OpenFOAM-2.0.0 and Paraview-3.10.1 are now installed in the */opt* directory.

OpenCFD社のHP
http://www.openfoam.com/

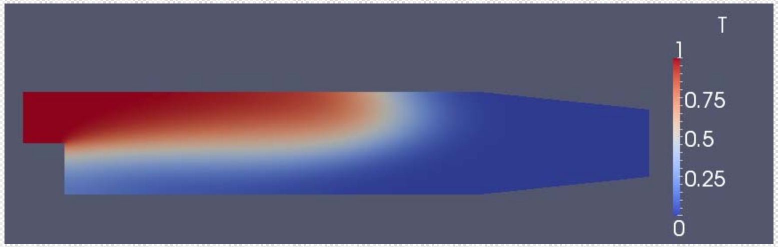
scalarTransportFoam

既に計算済みの風の場に、風の流れに追従する濃度物質をのせて、濃度物質の移流拡散を見るソルバー

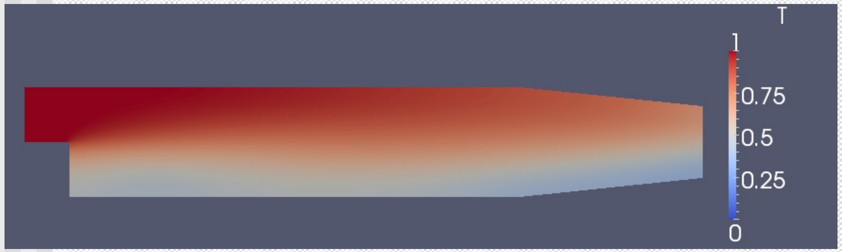
《 pitzDaily 》



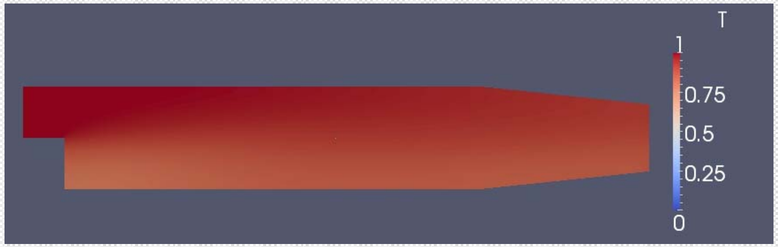
t=0



t=0.025

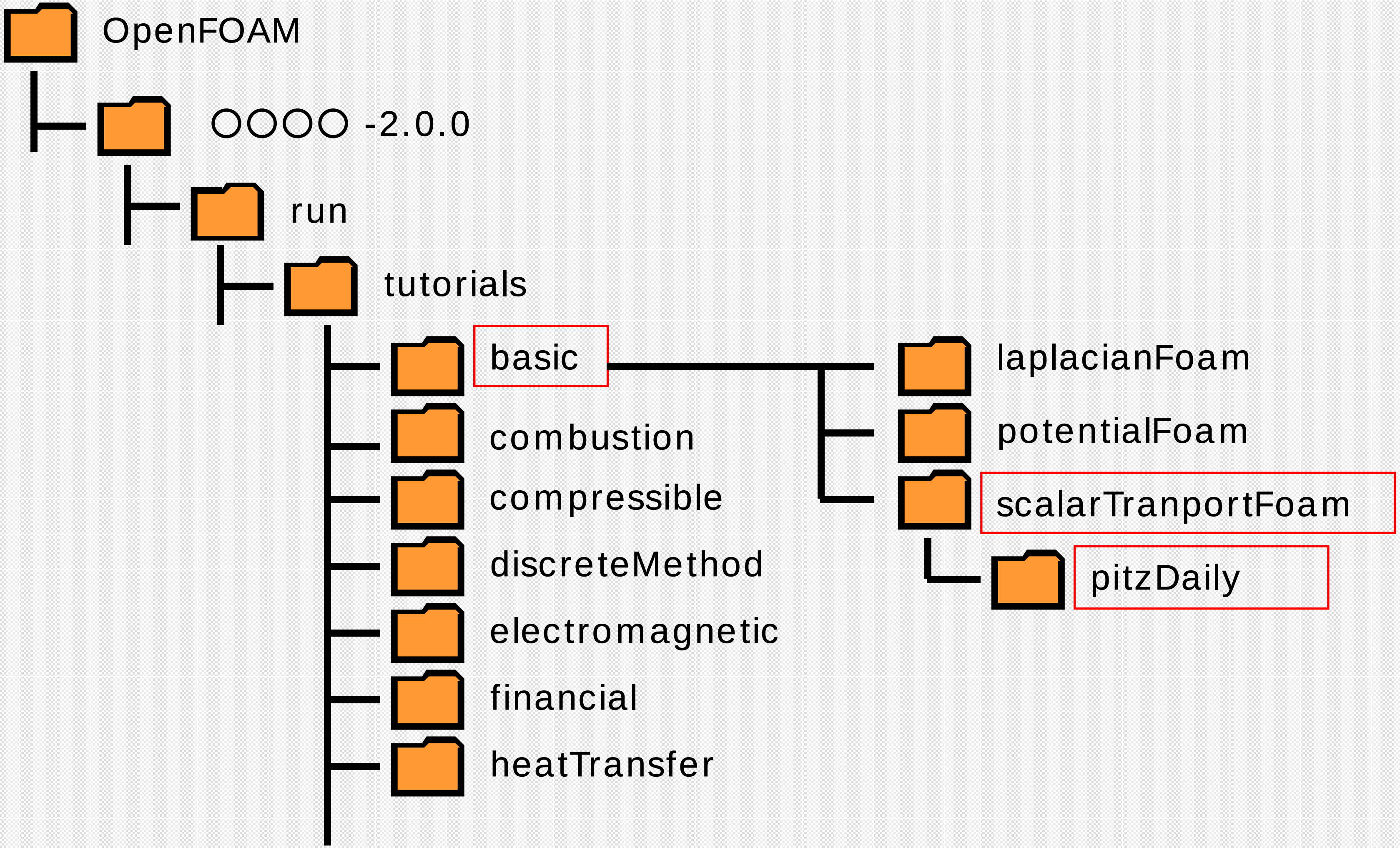


t=0.05



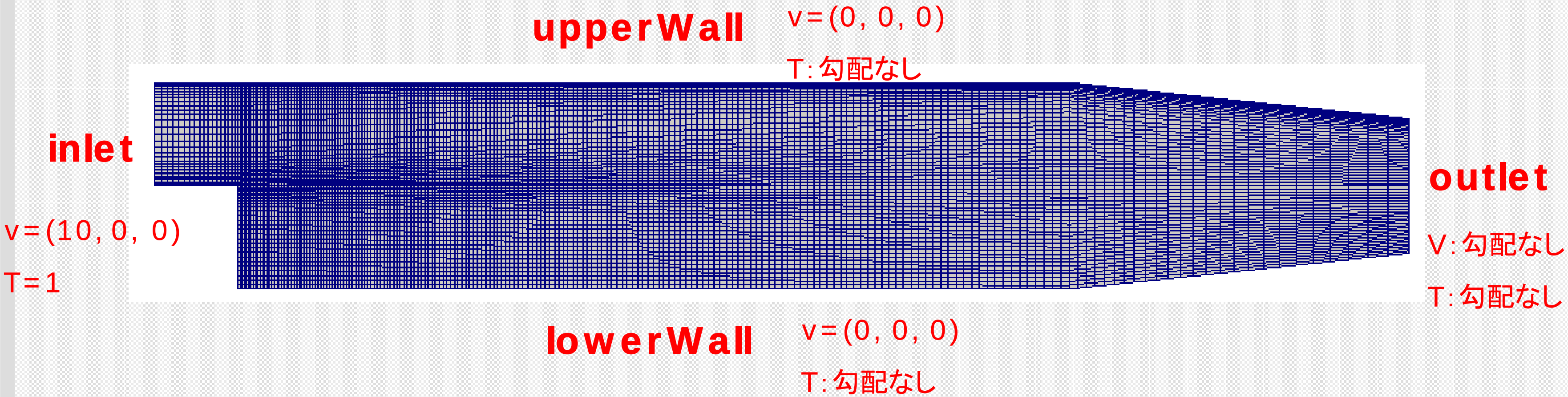
t=0.1

scalarTransportFoam/pitzDailyのチュートリアルの場所



scalarTransportFoam/pitzDailyの計算条件

項目	内容
①ソルバー	scalarTransportFoam
②支配方程式	$\frac{\partial T}{\partial t} + v \cdot \nabla T - D \nabla^2 T = 0$
③計算時間	0.1[s]
④計算ステップ	0.0001[s]
⑤境界条件	下記



Caseファイルの構造

- pitzDaily
  - system
    - controlDict　時間に関する条件
    - fvSchemes　計算スキーム
    - fvSolution　許容誤差
  - constant
    - transportProperties　輸送に関する物理パラメーター
    - polyMesh
      - blockMeshDict　メッシュ生成
      - boundary
  - 0　変数の初期条件
    - U
    - T

system/controlDict

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     scalarTransportFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         0.1;

deltaT          0.0001;

writeControl    timeStep;

writeInterval   50;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;
```

ソルバー

開始時間

終了時間

時間ステップ

何ステップ毎に出力ファイルを生成するか

system/ fvSchemes

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;                 時間スキーム
}

gradSchemes
{
    default         Gauss linear;          勾配スキーム
}

divSchemes
{
    default         none;                  発散スキーム
    div(phi,T)       Gauss limitedLinear 1;
}

laplacianSchemes
{
    default         none;                  ラプラシアンスキーム
    laplacian(DT,T) Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;                補間スキーム
}
```

```
snGradSchemes
{
    default         corrected;             表面法線方向勾配スキーム
}

fluxRequired
{
    default         no;                    流束の算出
    T               ;
}
```

system/ fvSolution

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    T
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-06;
        relTol          0;
    }
}

SIMPLE
{
    nNonOrthogonalCorrectors  0;
}
```

何ステップ毎に出力ファイルを生成するか

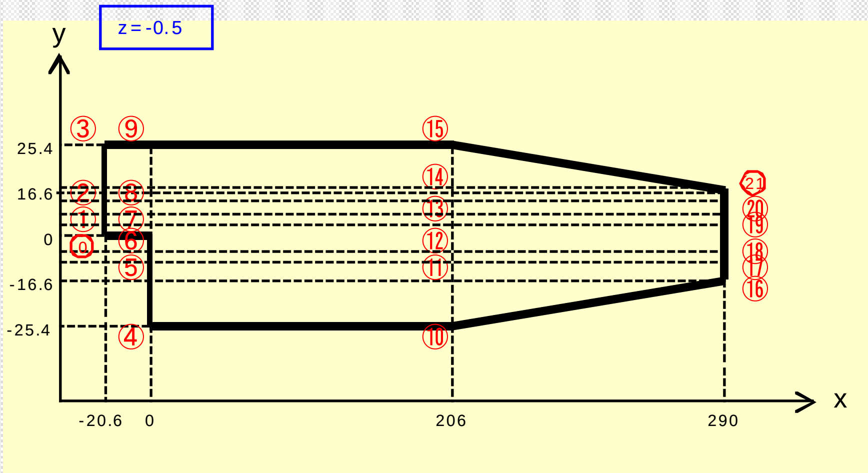
control/polyMesh/blockMeshDict

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * *   * * * * * //

convertToMeters 0.001;

vertices
(
    (-20.6 0 -0.5)         ⓪
    (-20.6 3 -0.5)         ①
    (-20.6 12.7 -0.5)      ②
    (-20.6 25.4 -0.5)      ③
    (0 -25.4 -0.5)         ④
    (0 -5 -0.5)            ⑤
    (0 0 -0.5)             ⑥
    (0 3 -0.5)             ⑦
    (0 12.7 -0.5)          ⑧
    (0 25.4 -0.5)          ⑨
    (206 -25.4 -0.5)       ⑩
    (206 -8.5 -0.5)        ⑪
    (206 0 -0.5)           ⑫
    (206 6.5 -0.5)         ⑬
    (206 17 -0.5)          ⑭
    (206 25.4 -0.5)        ⑮
    (290 -16.6 -0.5)       ⑯
    (290 -6.3 -0.5)        ⑰
    (290 0 -0.5)           ⑱
    (290 4.5 -0.5)         ⑲
    (290 11 -0.5)          ⑳
    (290 16.6 -0.5)        ㉑
    (-20.6 0 0.5)
    (-20.6 3 0.5)
    (-20.6 12.7 0.5)
    (-20.6 25.4 0.5)
    (0 -25.4 0.5)
    (0 -5 0.5)
    (0 0 0.5)
    (0 3 0.5)
    (0 12.7 0.5)
    (0 25.4 0.5)
    (206 -25.4 0.5)
    (206 -8.5 0.5)
    (206 0 0.5)
    (206 6.5 0.5)
    (206 17 0.5)
    (206 25.4 0.5)
    (290 -16.6 0.5)
    (290 -6.3 0.5)
    (290 0 0.5)
    (290 4.5 0.5)
    (290 11 0.5)
    (290 16.6 0.5)
);
```

頂点の定義



```
blocks        六面体の頂点    メッシュ数             倍数比率
(
    hex (0 6 7 1 22 28 29 23) (18 7 1) simpleGrading (0.5 1.8 1)
    hex (1 7 8 2 23 29 30 24) (18 10 1) simpleGrading (0.5 4 1)
    hex (2 8 9 3 24 30 31 25) (18 13 1) simpleGrading (0.5 0.25 1)
    hex (4 10 11 5 26 32 33 27) (180 18 1) simpleGrading (4 1 1)
    hex (5 11 12 6 27 33 34 28) (180 9 1) edgeGrading (4 4 4 4 0.5 1 1 0.5 1 1 1 1)
    hex (6 12 13 7 28 34 35 29) (180 7 1) edgeGrading (4 4 4 4 1.8 1 1 1.8 1 1 1 1)
    hex (7 13 14 8 29 35 36 30) (180 10 1) edgeGrading (4 4 4 4 4 1 1 4 1 1 1 1)
    hex (8 14 15 9 30 36 37 31) (180 13 1) simpleGrading (4 0.25 1)
    hex (10 16 17 11 32 38 39 33) (25 18 1) simpleGrading (2.5 1 1)
    hex (11 17 18 12 33 39 40 34) (25 9 1) simpleGrading (2.5 1 1)
    hex (12 18 19 13 34 40 41 35) (25 7 1) simpleGrading (2.5 1 1)
    hex (13 19 20 14 35 41 42 36) (25 10 1) simpleGrading (2.5 1 1)
    hex (14 20 21 15 36 42 43 37) (25 13 1) simpleGrading (2.5 0.25 1)
);

edges
(
);

boundary   境界条件
(
    inlet
    {
        type patch;
        faces
        (
            (0 22 23 1)        面（四角形）の頂点
            (1 23 24 2)
            (2 24 25 3)
        );
    }
    outlet
    {
        type patch;
        faces
        (
            (16 17 39 38)
            (17 18 40 39)
            (18 19 41 40)
            (19 20 42 41)
            (20 21 43 42)
        );
    }
```

```
upperWall
    {
        type wall;
        faces
        (
            (3 25 31 9)
            (9 31 37 15)
            (15 37 43 21)
        );
    }
    lowerWall
    {
        type wall;
        faces
        (
            (0 6 28 22)
            (6 5 27 28)
            (5 4 26 27)
            (4 10 32 26)
            (10 16 38 32)
        );
    }
    frontAndBack
    {
        type empty;
        faces
        (
            (22 28 29 23)
            (23 29 30 24)
            (24 30 31 25)
            (26 32 33 27)
            (27 33 34 28)
            (28 34 35 29)
            (29 35 36 30)
            (30 36 37 31)
            (32 38 39 33)
            (33 39 40 34)
            (34 40 41 35)
            (35 41 42 36)
            (36 42 43 37)
```

z方向の境界条件の定義

(2次元の計算でもz方向に1セル設けている)

```
            (0 1 7 6)
            (1 2 8 7)
            (2 3 9 8)
            (4 5 11 10)
            (5 6 12 11)
            (6 7 13 12)
            (7 8 14 13)
            (8 9 15 14)
            (10 11 17 16)
            (11 12 18 17)
            (12 13 19 18)
            (13 14 20 19)
            (14 15 21 20)
        );
    }
);

mergePatchPairs
(
);
```

control/ transportProperties

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * //

DT              DT [ 0 2 -1 0 0 0 0 ] 0.01;
```

拡散係数

0/ v

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

dimensions      [0 1 -1 0 0 0 0];

internalField   nonuniform List<vector>
12225
(
(9.88226 -1.12989 2.24499e-47)
(9.78836 -0.592567 3.02929e-46)
...
(4.04216 -0.408617 -4.48223e-20)
)
;

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform (10 0 0);
    }

    outlet
    {
        type            zeroGradient;
    }
```

セルごとの初期値

（本ケースの場合は既に解かれたものが与えられている）

境界条件

```
    upperWall
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

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

    frontAndBack
    {
        type            empty;
    }
}
```

0/ T

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      T;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 1 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform 1;
    }

    outlet
    {
        type            zeroGradient;
    }

    upperWall
    {
        type            zeroGradient;
    }
    lowerWall
    {
        type            zeroGradient;
    }

    frontAndBack
    {
        type            empty;
    }
}
```

セルごとの初期値

（濃度については一様0）

境界条件

/opt/openfoam200/applications/solvers/basic/scalarTranportFoam/scalarTransportFoam.C

```
#include "fvCFD.H"
#include "simpleControl.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * ** * * //

int main(int argc, char *argv[])
{
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "createFields.H"

    simpleControl simple(mesh);

    // * * * * * * * * * * * * * * * * * * * * * * * * * //

    Info<< "¥nCalculating scalar transport¥n" << endl;

    #include "CourantNo.H"

    while (simple.loop())
    {
        Info<< "Time = " << runTime.timeName() << nl << endl;

        for (int nonOrth=0; nonOrth<=simple.nNonOrthCorr(); nonOrth++)
        {
            solve
            (
                fvm::ddt(T)
              + fvm::div(phi, T)
              - fvm::laplacian(DT, T)
            );
        }

        runTime.write();
    }
```

$$\Leftrightarrow \quad \frac{\partial T}{\partial t} + v \cdot \nabla T - D \nabla^2 T = 0$$

```
    Info<< "End¥n" << endl;

    return 0;
}
```

数式を変更するには新たにソルバーを作り直さないとならない

計算実行

メッシュ生成

```
blockMesh
```

blockMesh とsnappyHexMesh

まずblockMeshを実行し

更にsystemフォルダに
snappyHexMeshDictがあれば
snappyHexMeshを実行

計算実行

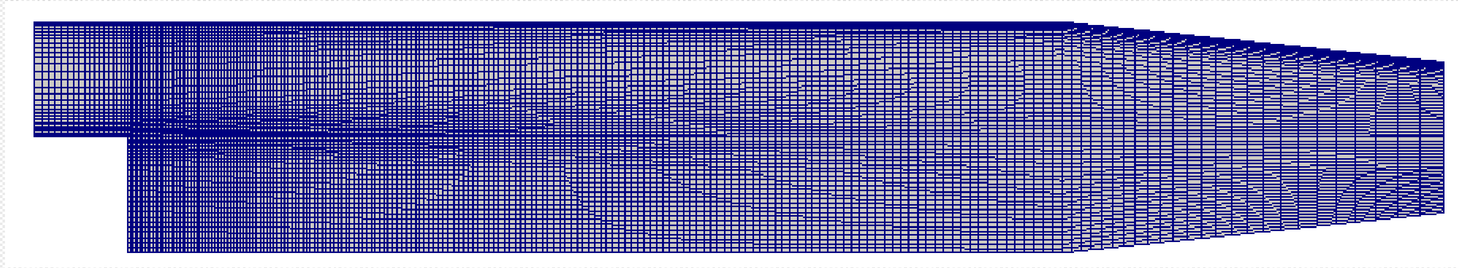
```
scalarTransportFoam
```

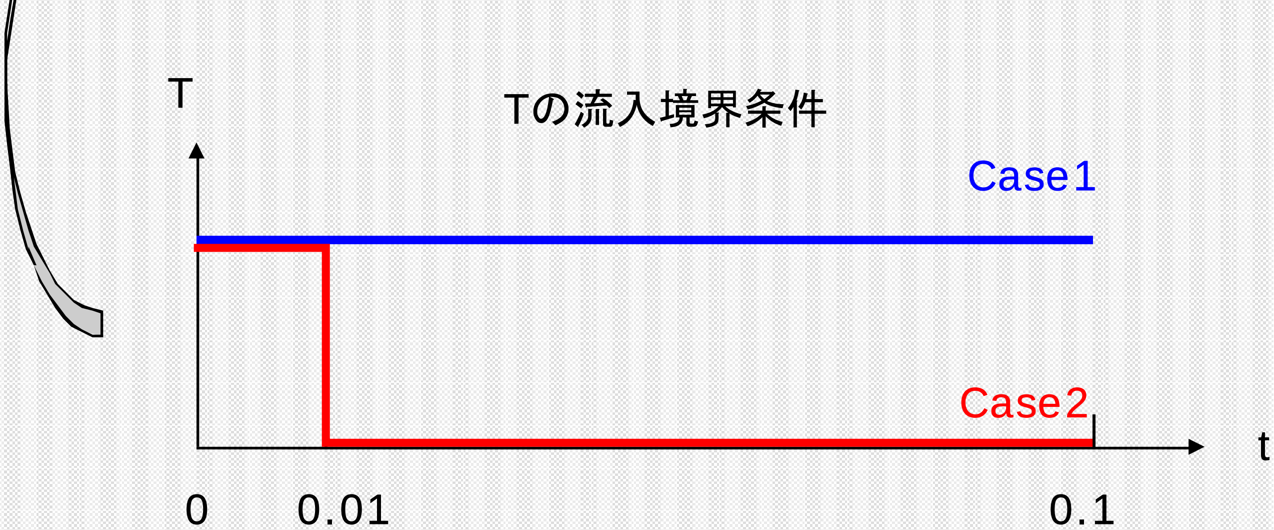
controlDictのapplicationに書いてあるソルバー名を入力してリターン

可視化

```
paraFoam
```

応用：境界条件を少し変えてみる





0/T

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      T;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 1 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
//        type            fixedValue;
//        value           uniform 1;
        type            timeVaryingUniformFixedValue;
                fileName        "0/T.dat";
                outOfBounds                     clump;
    }
```

0/T.dat

```
(
( 0 1 )
( 0.01 0 )
)
```

0[s]でT=1 そのまま続ける
0.01[s]でT=0 そのまま続ける