実践ソルバー改良①
scalarTransportFoam の改良事例

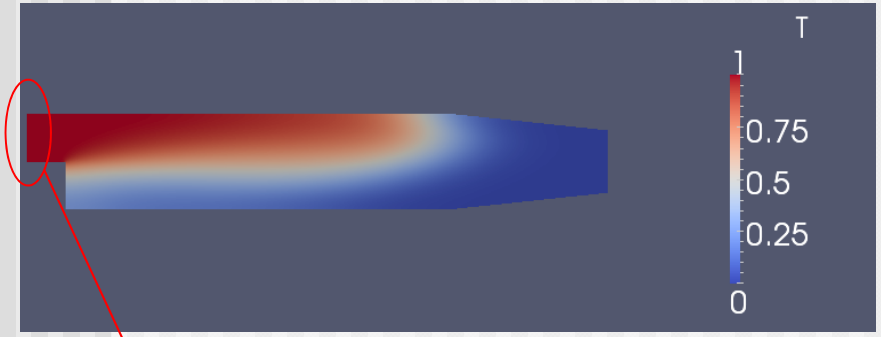
Contents:

柴田貴裕

1 ねらい

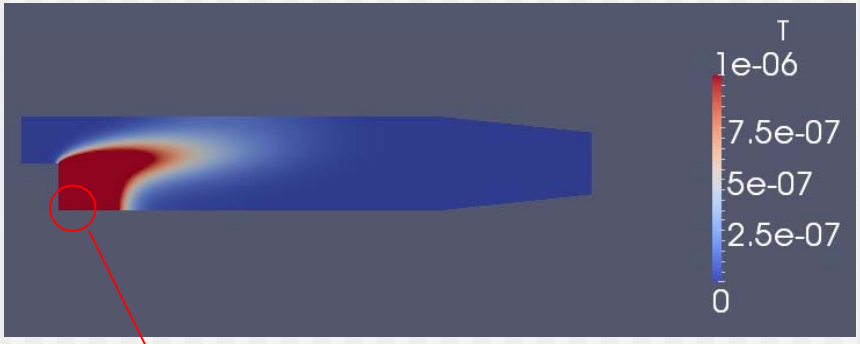
Passivescalar粒子の移流拡散に関して、既存のTutorialケースを元にソルバーのカスタマイズを行う

既存のチュートリアル



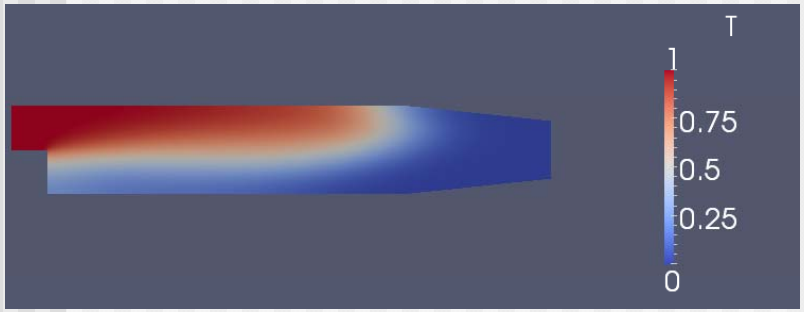
拡散する物質が境界から流入

今回の課題



拡散する物質が左下隅の1セルから発生

2 既存のモデルと改造のポイント



① tutorials/basic/scalarTransportFoam のケースファイルを改造する

② このケースファイルは与えられた風にpassivescalar粒子を乗せ、粒子の濃度についての移流拡散方程式を解く

③ 既存のソルバーには発生項がないので、発生項入りのソルバーを作る

$$\frac{\partial T}{\partial t} + v \cdot \nabla T = D\nabla^2 T + S$$

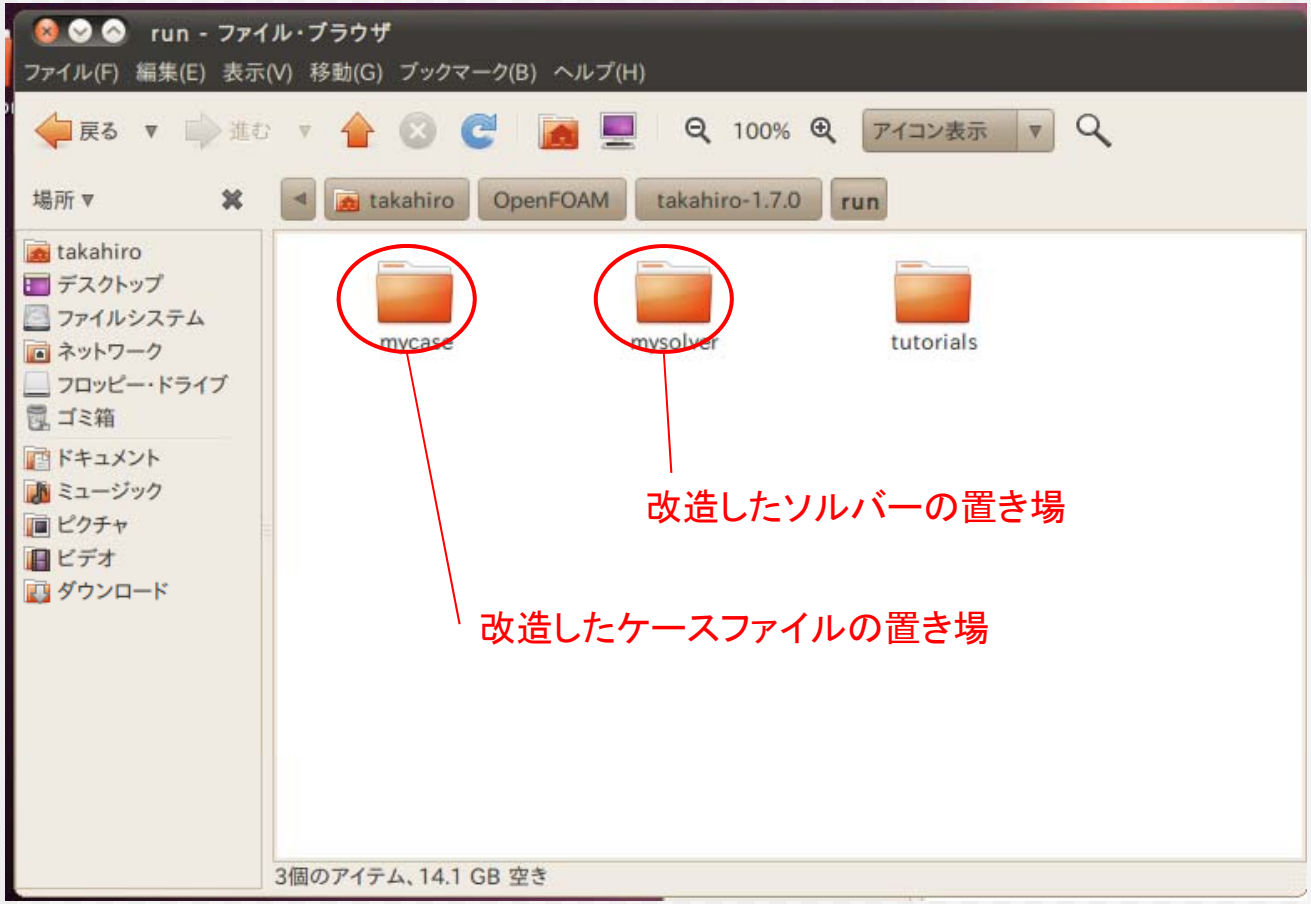
発生項

T:粒子濃度　　v:風速
D:拡散係数　　S:粒子発生量

3 ソルバーの改造

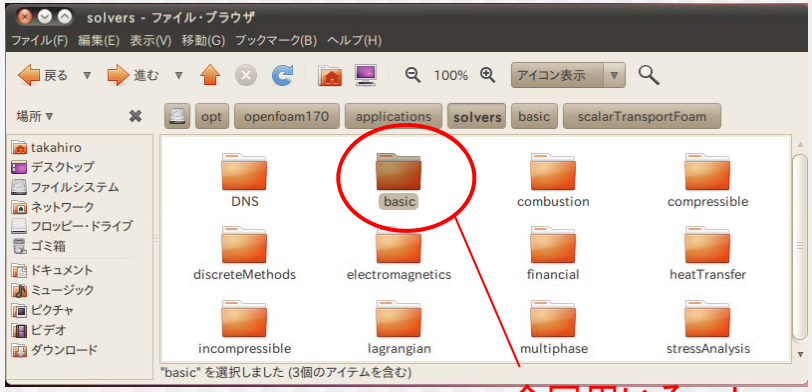
Step1 ソルバー & ケースファイルのコピー

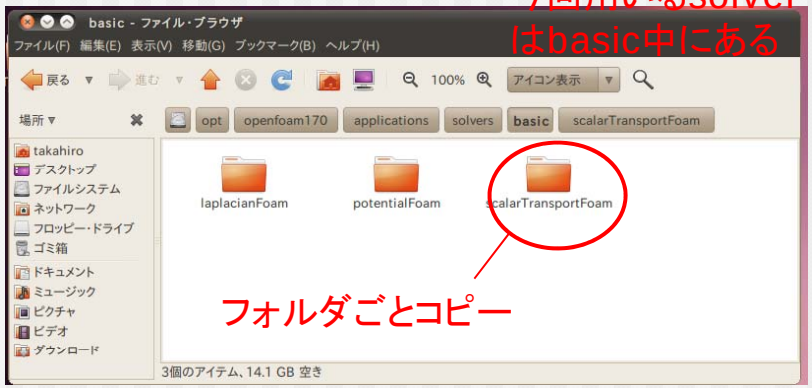
① コピー先のフォルダの作成

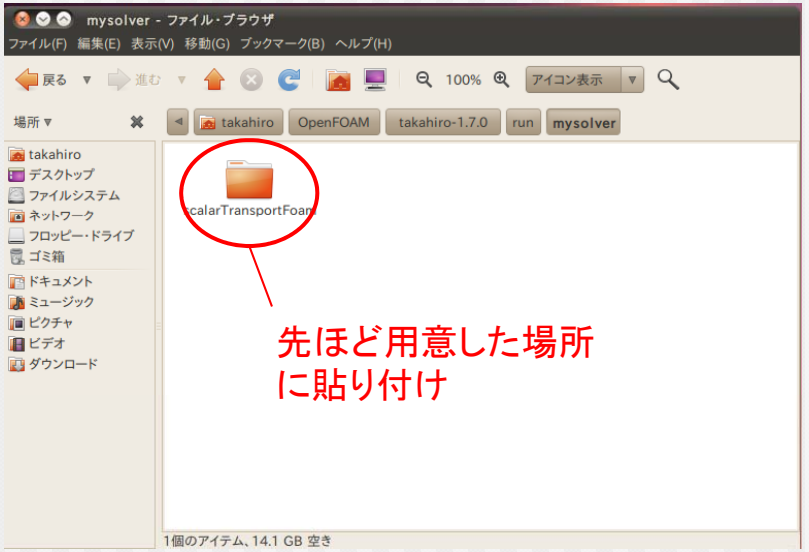


Step1 ソルバー & ケースファイルのコピー

② 既存ソルバーのコピー & ペースト

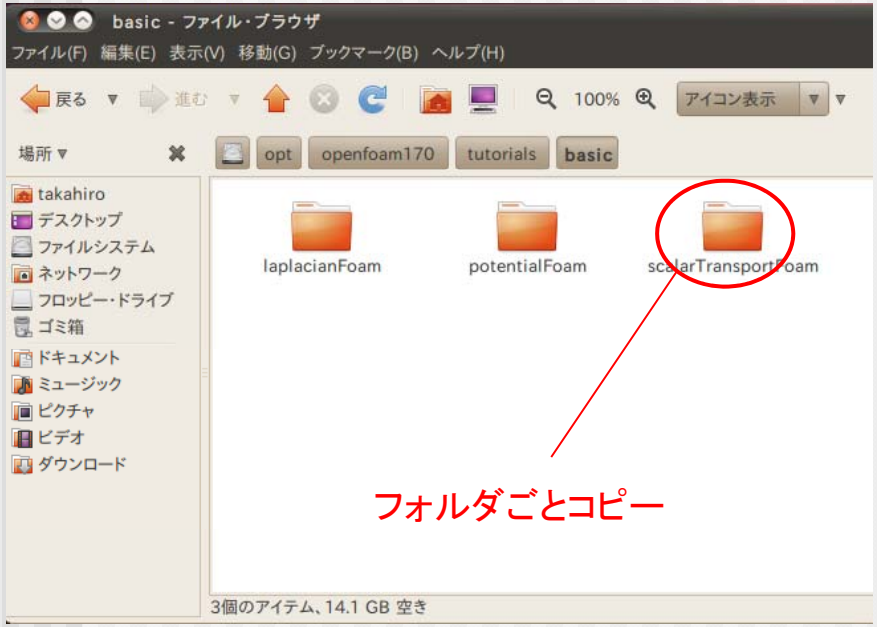


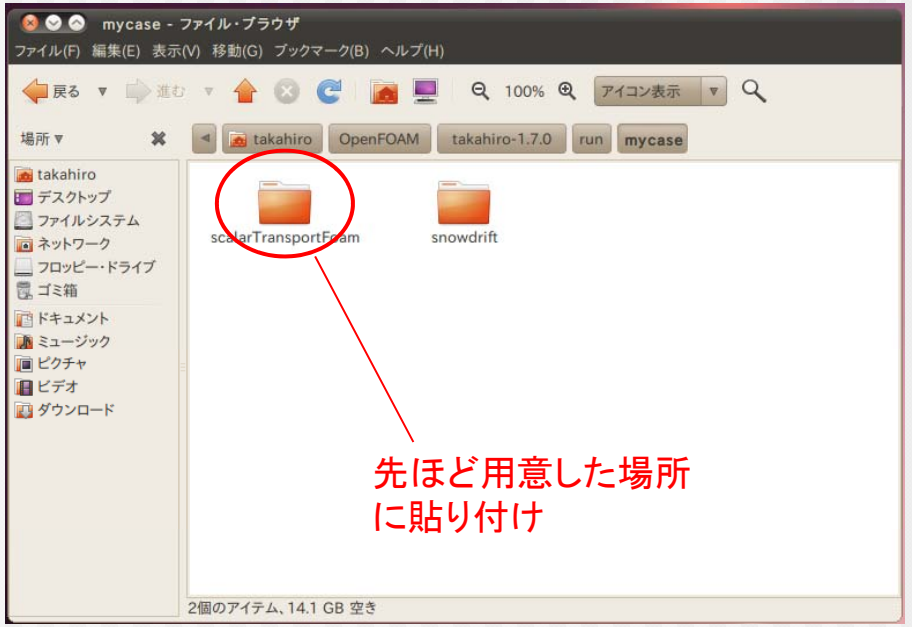




Step1 ソルバー & ケースファイルのコピー

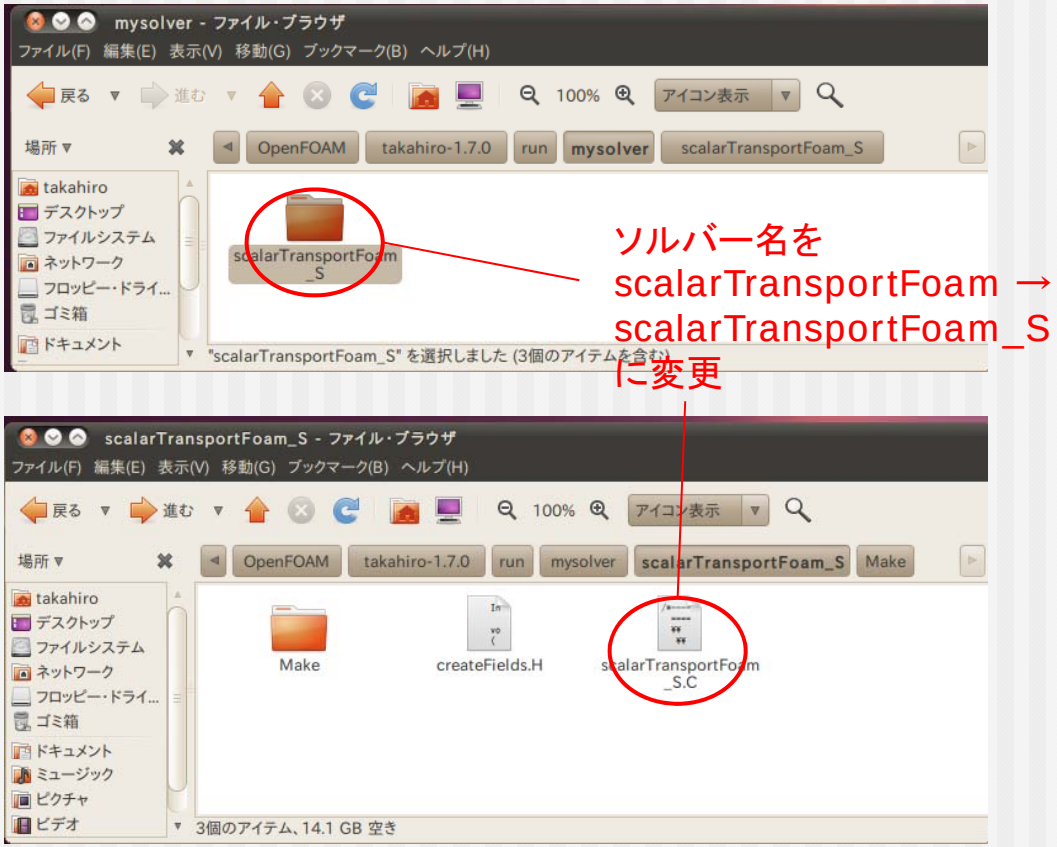
③ 既存ケースファイルのコピー & ペースト

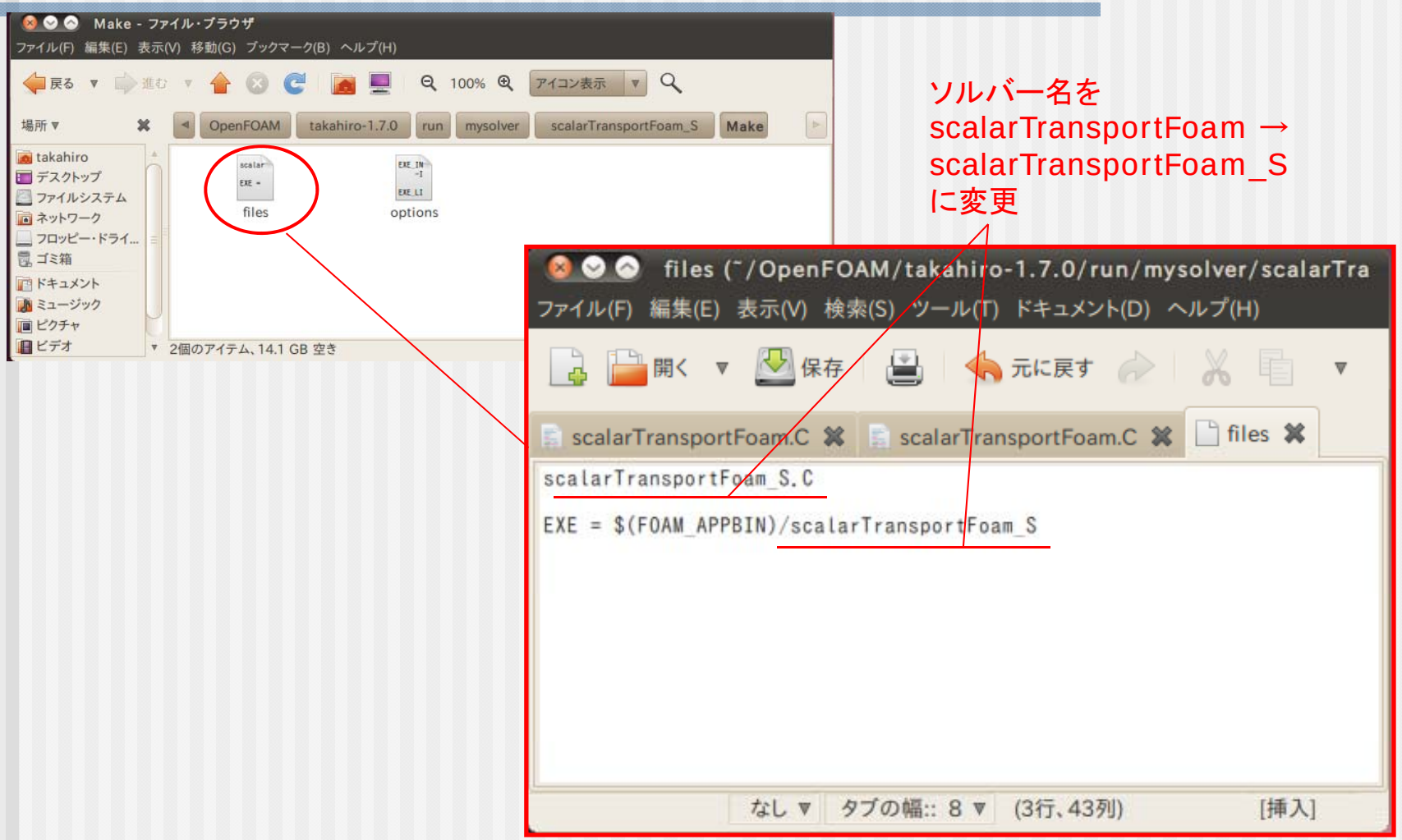




Step2 ソルバー改造

① ソルバーの名前の変更




Make - ファイル・ブラウザ
ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ヘルプ(H)
戻る
進む
100%
アイコン表示
場所
OpenFOAM
takahiro-1.7.0
run
mysolver
scalarTransportFoam_S
Make
takahiro
デスクトップ
ファイルシステム
ネットワーク
フロッピー・ドライ...
ゴミ箱
ドキュメント
ミュージック
ピクチャ
ビデオ
files
options
2個のアイテム、14.1 GB 空き
ソルバー名を
scalarTransportFoam →
scalarTransportFoam_S
に変更
files (~/OpenFOAM/takahiro-1.7.0/run/mysolver/scalarTra
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
scalarTransportFoam.C
scalarTransportFoam.C
files
scalarTransportFoam_S.C
EXE = $(FOAM_APPBIN)/scalarTransportFoam_S
なし
タブの幅: 8
(3行、43列)
[挿入]

Step2 ソルバー改造

② 輸送方程式に発生項を加える

scalarTranportFoam_S.C

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

#       include "readSIMPLEControls.H"

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

        runTime.write();
    }
```

この項を追加

$$\frac{\partial T}{\partial t} + v \cdot \nabla T = D\nabla^2 T + S$$

発生項

Step2 ソルバー改造

③ 発生項変数Sに関する処理を
createFields.Hで定義

createFields.H

```
Info<< "Reading field T¥n" << endl;

volScalarField T
(
    IOobject
    (
        "T",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

コピーして変数T
をSに変更する

```
Info<< "Reading field S¥n" << endl;

volScalarField S
(
    IOobject
    (
        "S"
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

ここを変更する

```
Info<< "Reading field U¥n" << endl;
```

Step2 ソルバー改造

③ ソルバーのコンパイル

ターミナルを立ち上げ先ほど改変したソルバーのフォルダに移動

```
cd OpenFOAM/takahiro-1.7.0/run/mysolver/scalarTransportFoam_S/
```

コンパイル

```
wmake
```

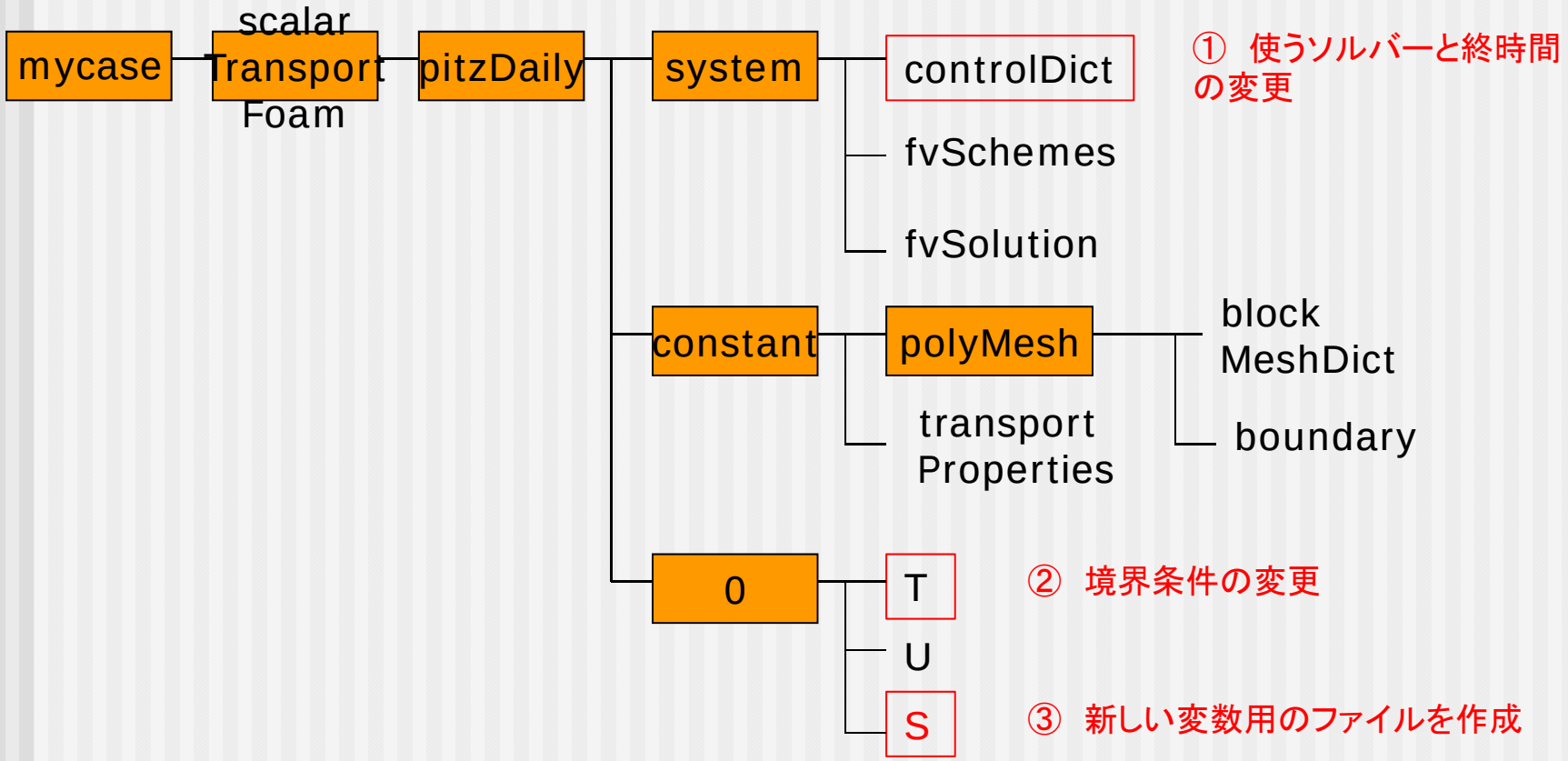
もし、permission deniedが出たら、chownで所有者を変えた後、wmake

```
sudo chown -R takahiro /opt/openfoam170/applications/
```

所有者

Step3 ケースファイル改造

ケースファイルの変更点



① controlDict

変更前

```
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 uncompressed;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;
```

変更後

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

application     scalarTransportFoam_S;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         0.01;

deltaT          0.0001;

writeControl    timeStep;

writeInterval   50;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression uncompressed;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;
```

② 0/T

変更前

```
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;
    }
}

// ************************************************************************* //
```

変更後

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

dimensions      [0 0 0 1 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }

    outlet
    {
        type            zeroGradient;
    }

    upperWall
    {
        type            zeroGradient;
    }

    lowerWall
    {
        type            zeroGradient;
    }

    frontAndBack
    {
        type            empty;
    }
}

// ************************************************************************* //
```

③ 0/S

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      S;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

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

internalField   nonuniform List<scalar>
12225
(
0
0
0
0
0
0
0
0
0
⋮
0
0
0
1
0
0
0
0
⋮
```

← 540行目に1が出てくる

```
⋮
0
0
0
)
;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }
    outlet
    {
        type            zeroGradient;
    }
    upperWall
    {
        type            zeroGradient;
    }
    lowerWall
    {
        type            zeroGradient;
    }
    frontAndBack
    {
        type            empty;
    }
}
```

Step4 実行

メッシュ生成

```
blockMesh
```

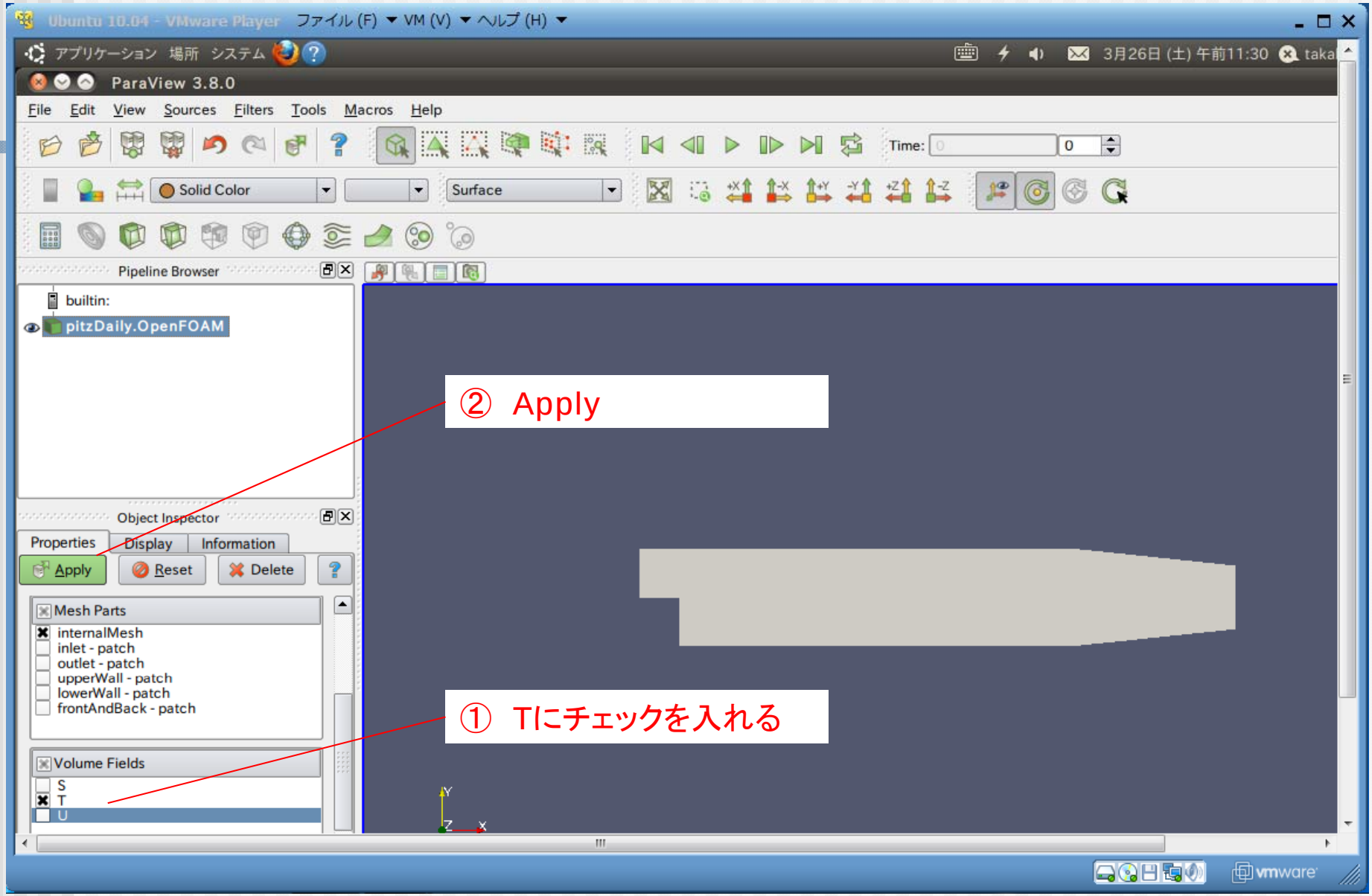
計算の実行

```
scalarTransportFoam_S
```

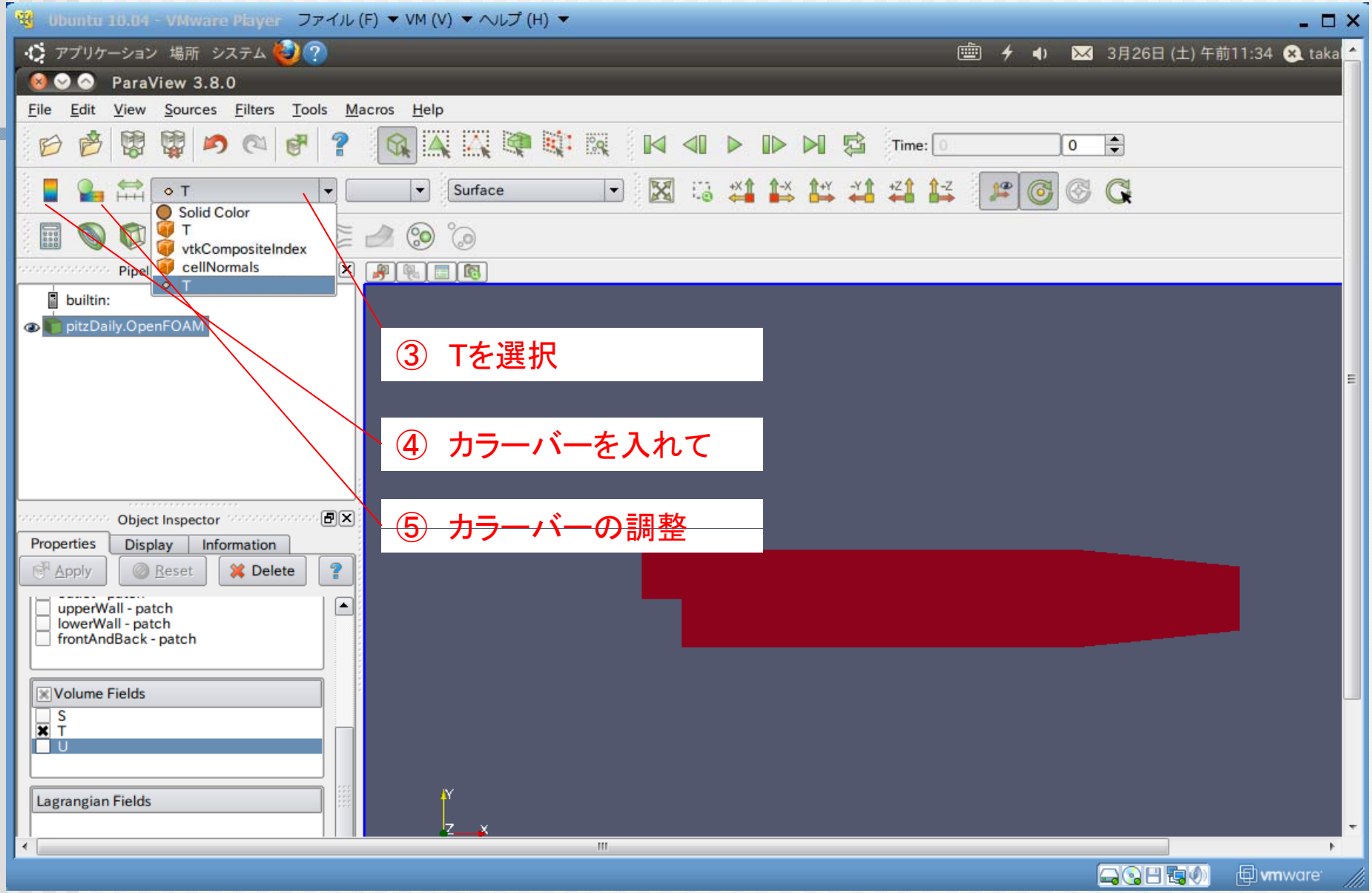
結果の可視化

```
paraFoam
```

paraview

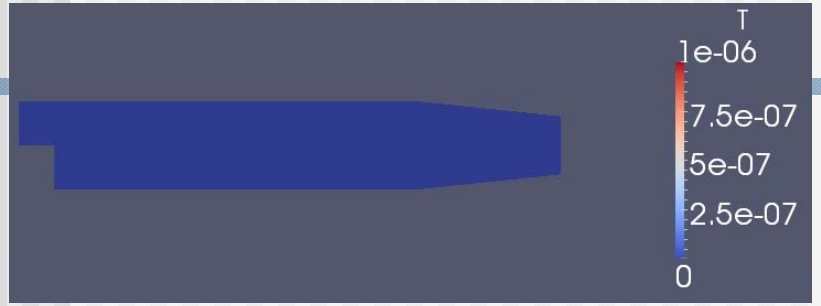

Ubuntu 10.04 - VMware Player
ファイル (F)
VM (V)
ヘルプ (H)
アプリケーション
場所
システム
3月26日 (土) 午前11:30
taka
ParaView 3.8.0
File
Edit
View
Sources
Filters
Tools
Macros
Help
Time:
Solid Color
Surface
Pipeline Browser
builtin:
pitzDaily.OpenFOAM
Object Inspector
Properties
Display
Information
Apply
Reset
Delete
Mesh Parts
internalMesh
inlet - patch
outlet - patch
upperWall - patch
lowerWall - patch
frontAndBack - patch
Volume Fields
S
T
U
② Apply
① Tにチェックを入れる
vmware

paraview

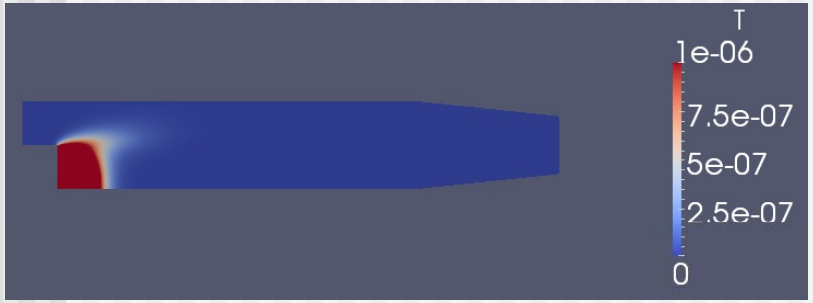

Ubuntu 10.04 - VMware Player
ファイル (F)
VM (V)
ヘルプ (H)
アプリケーション
場所
システム
3月26日 (土) 午前11:34
ParaView 3.8.0
File
Edit
View
Sources
Filters
Tools
Macros
Help
Time:
Surface
Solid Color
T
vtkCompositeIndex
cellNormals
builtin:
pitzDaily.OpenFOAM
③ Tを選択
④ カラーバーを入れて
⑤ カラーバーの調整
Object Inspector
Properties
Display
Information
Apply
Reset
Delete
upperWall - patch
lowerWall - patch
frontAndBack - patch
Volume Fields
S
T
U
Lagrangian Fields
vmware

paraview

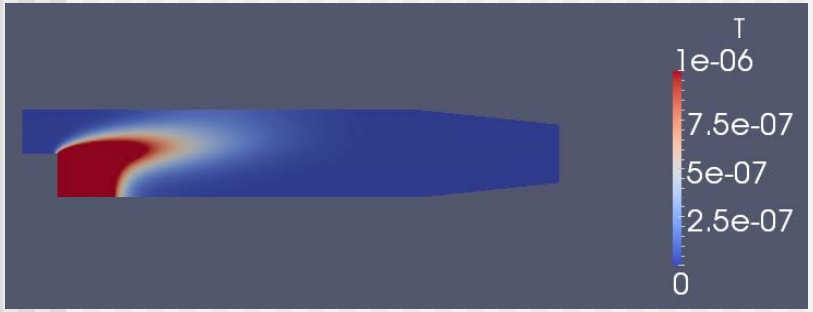
t=0



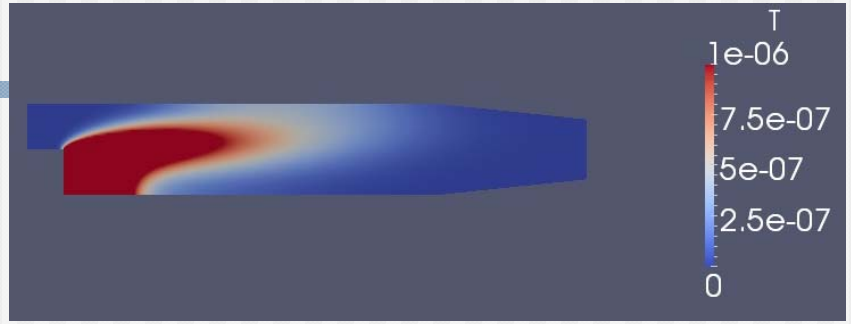
t=0.01



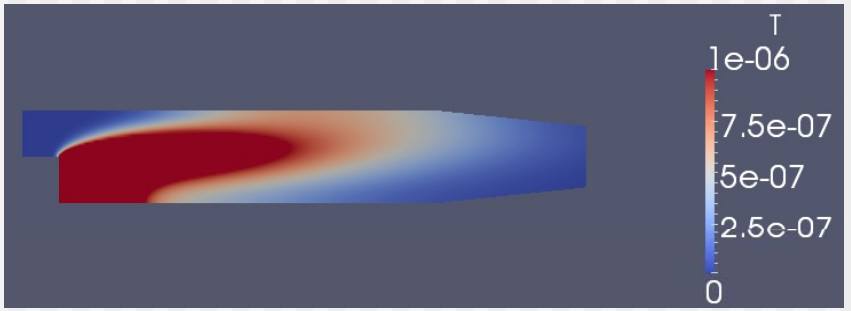
t=0.02



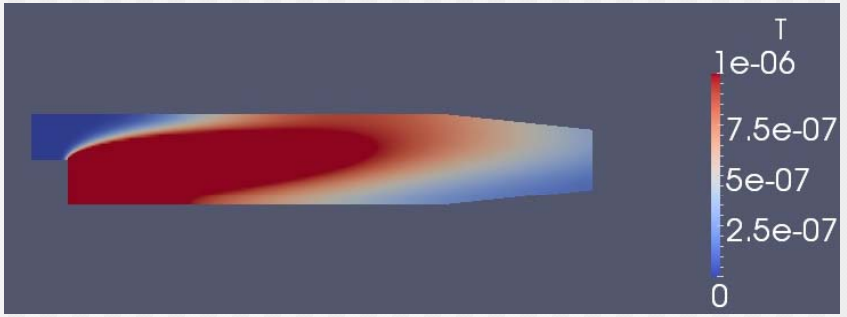
t=0.03



t=0.04



t=0.05



Appendix　粒子発生部の指定方法

実演では粒子発生部に発生量が記載された変数Sファイルを用いたが、このファイルはsetFieldsユーティティを用いて作成することができる。

Systemフォルダの下に、下記のsetFieldsDictを置く

setFieldsDict

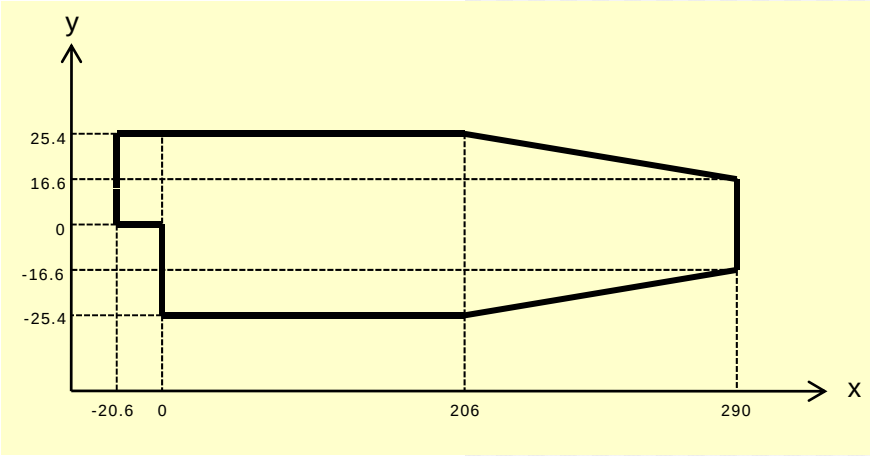
```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      setFieldsDict;
}

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

defaultFieldValues
(
    volScalarFieldValue S 0
);

regions
(
    nearestToCell
    {
        points ((0.000001 -0.025399 0));

        fieldValues
        (
            volScalarFieldValue S 1
        );
    }
);
```

変数Sの全てのセル値を0にする

指定した点に最も近いセルを選択

指定の点

変数Sの指定セル値を1にする



変数Sファイルの雛型を準備

変数Tファイルをコピーして、下記の変更を行い名前をSに変えて保存

変更前　0/T

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

dimensions      [0 0 0 1 0 0 0]; ← 単位[K]

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }

    outlet
    {
        type            zeroGradient;
    }

    upperWall
    {
        type            zeroGradient;
    }

    lowerWall
    {
        type            zeroGradient;
    }

    frontAndBack
    {
        type            empty;
    }
}

// ************************************************************************* //
```

変更後 0/S

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

dimensions      [0 0 -1 1 0 0 0]; ← 単位[K/s]

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            zeroGradient;
    }

    outlet
    {
        type            zeroGradient;
    }

    upperWall
    {
        type            zeroGradient;
    }

    lowerWall
    {
        type            zeroGradient;
    }

    frontAndBack
    {
        type            empty;
    }
}

// ************************************************************************* //
```

ケースファイルの実行

メッシュ生成

```
blockMesh
```

変数Sの指定セルに発生量を書き込む

```
blockMesh
```

計算の実行

```
scalarTransportFoam_S
```

結果の可視化

```
paraFoam
```