OpenFoamのためのC/C++

第4回　ソルバーカスタマイズの基本

田中昭雄

目的

この勉強会の資料があれば、
OpenFoamカスタマイズ時にC/C++で迷わない

予定

- 第1回　メモリ管理
- 第2回　CFDの例で勉強するクラス
- 第3回　OpenFOAMで勉強するテンプレート
- 第4回　ソルバーカスタマイズの基本
- 第5回　デバッグ・ソースコード管理
- 第6回　未定(できればUtilitiesを使った何か)

今回のテーマ

ソルバーのカスタマイズ基本

Agenda

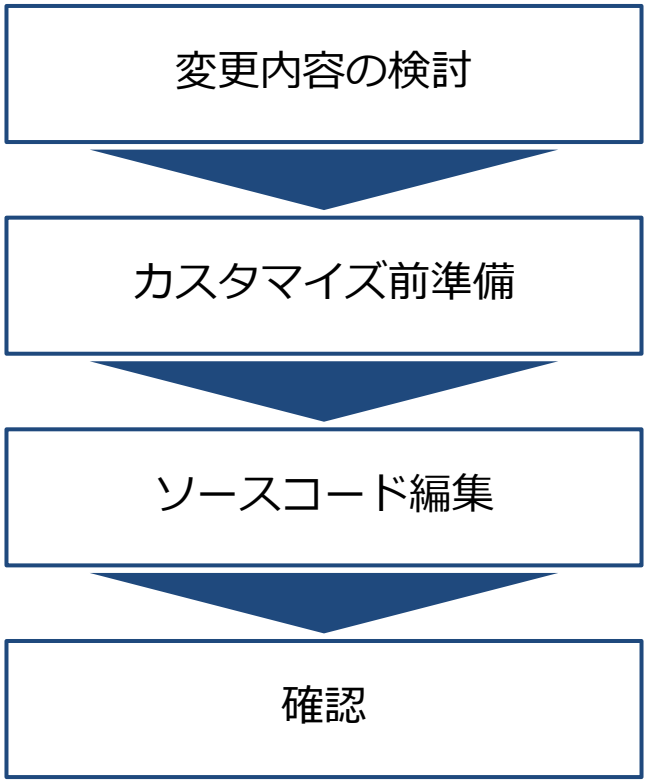
- ソルバーカスタマイズ手順概要
- ソルバーカスタマイズ例による手順詳細

Agenda

- ソルバーカスタマイズ手順概要
- ソルバーカスタマイズ例による手順詳細

カスタマイズ手順

既存機能をベースに機能変更 / 付加



カスタマイズ手順(1) - 変更内容の検討

実現したい事項の検討・決定

①カスタマイズ要望確認

②ベースソルバーの選択

カスタマイズ手順(2) – 前準備

既存コードを破壊しない / カスタマイズコードのビルド準備

①カスタマイズソルバーの名前を決定

②ベースソルバーのコピー

③ディレクトリ・ファイルの名前変更

④ビルドテスト
(コード変更なし・結果変わらない事確認)

カスタマイズ手順(3) – ソースコード編集

カスタマイズソルバー作成

①追加データ構造取り扱い処理追加
(設定ファイルから読込/出力など)

②カスタマイズ処理追加

③ビルドエラー削除

カスタマイズ手順(4) – 確認

要望どおりの機能実現されているか確認

①ケースファイルの作成

②カスタマイズソルバーの実行

③結果確認

Agenda

- ソルバーカスタマイズ手順概要
- ソルバーカスタマイズ例による手順詳細

カスタマイズ例

非定常層流・非圧縮性流体に
スカラー場を追加して非定常解析をしたい

変更内容の検討

icoFoamに対してスカラー場の計算を追加

①カスタマイズ要望確認

②ベースソルバーの選択

変更内容の検討

①カスタマイズ要望確認

対象とする流体

- 非定常層流
- 非圧縮性流体
- 非定常解析

追加するスカラー場の輸送方程式

$$\frac{\partial T}{\partial t}+\nabla\cdot\left(\mathbf{U}T\right)-\nabla\cdot\left(\nu\nabla T\right)=0$$

$\mathbf{U}$　流速　　ν　粘度

T　スカラー場（ex. 温度など）

変更内容の検討(2)

②ベースソルバーの選択

- 定常 / 非定常
- 層流 / 乱流
- 圧縮性 / 非圧縮性
- アルゴリズム

icoFoam利用
(非定常層流、非圧縮性、PISO)

カスタマイズ前準備

ベースとするソルバーのソースコードを用意

①カスタマイズソルバーの名前を決定

②ベースソルバーのコピー

③ディレクトリ・ファイルの名前変更

④ビルドテスト

カスタマイズ前準備

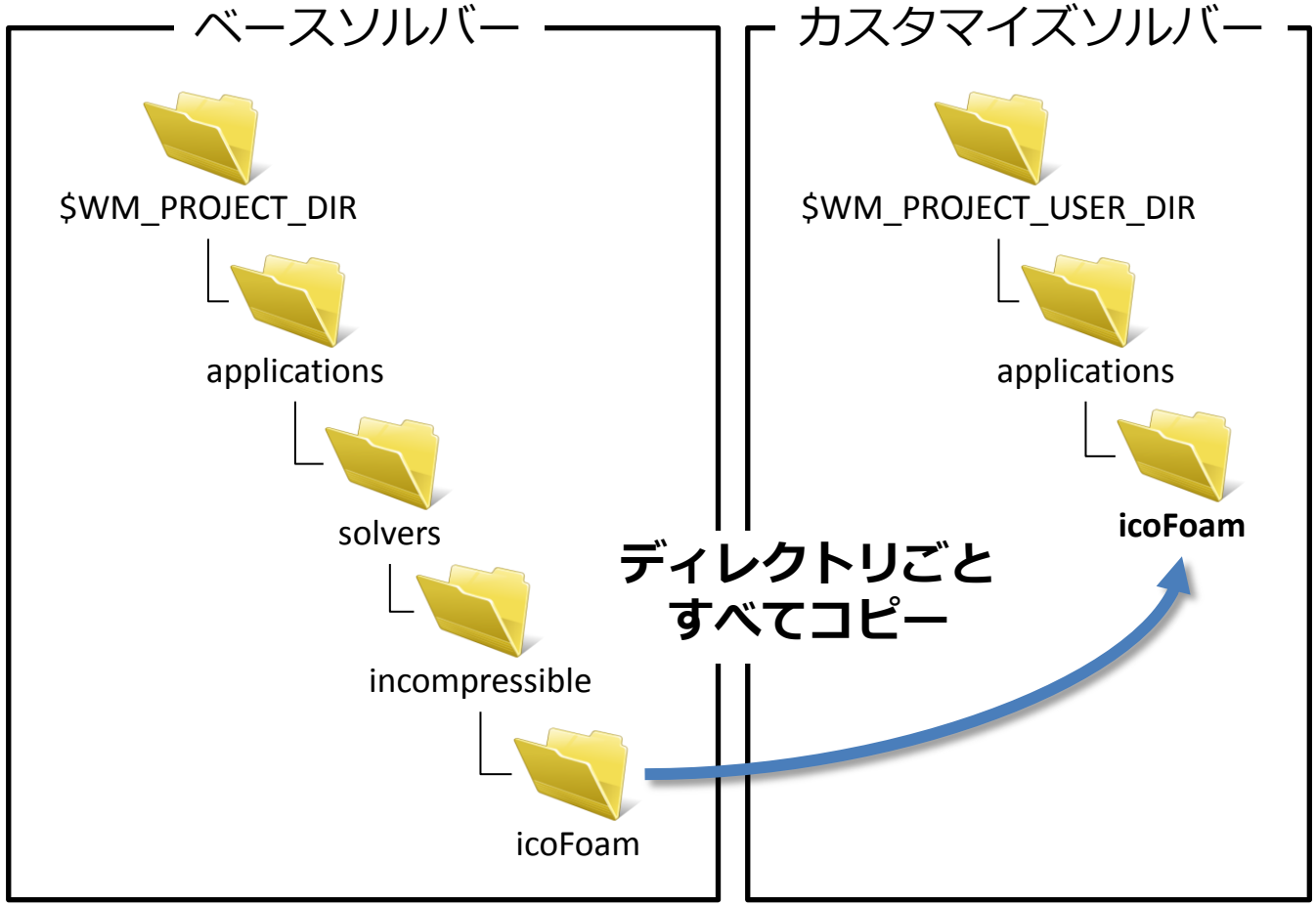
②ベースソルバーのコピー

変更内容の検討

カスタマイズ前準備

ソースコード編集

確認

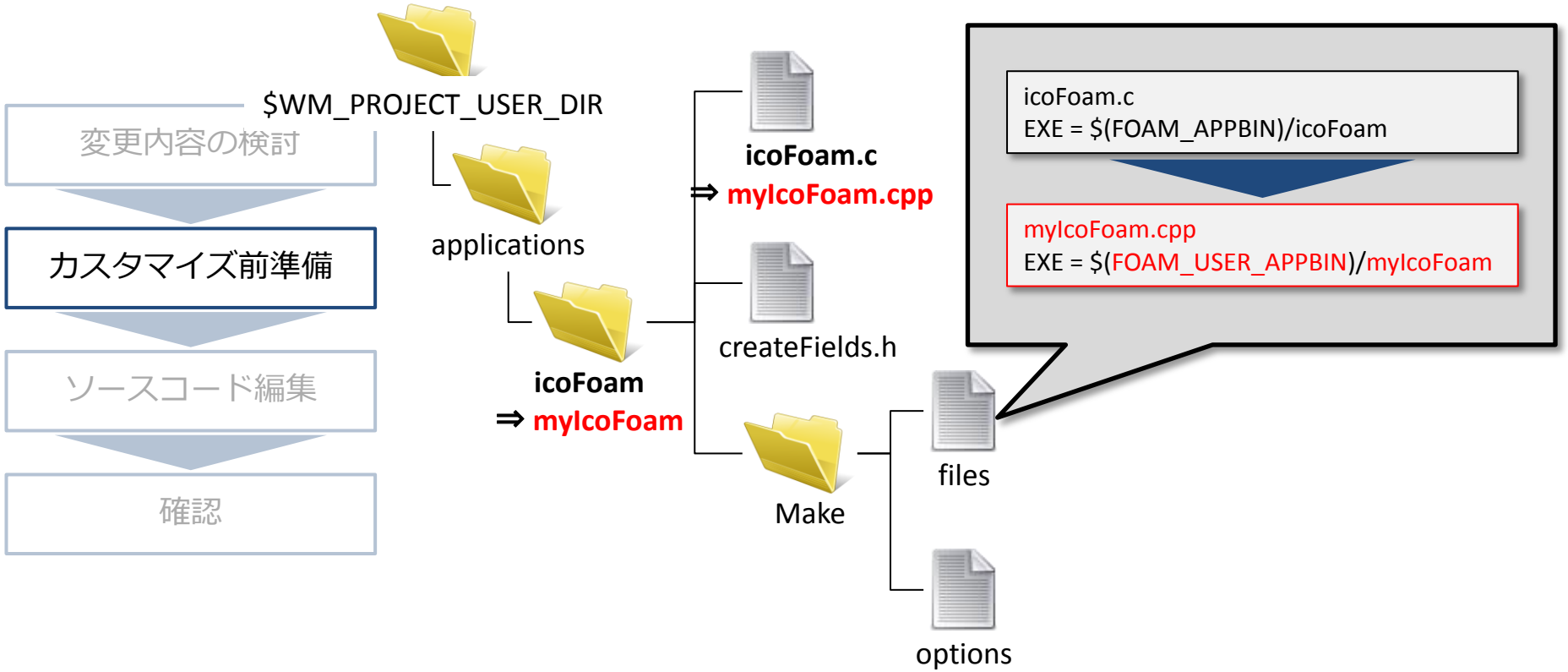


※シェルコマンド：

•ディレクトリ毎コピー：cp –r <コピー元ディレクトリパス> <コピー先ディレクトリパス>

カスタマイズ前準備

③ディレクトリ・ファイルの名前変更



※シェルコマンド：

•ディレクトリ・ファイル名変更：mv <変更前名前> <変更後名前>

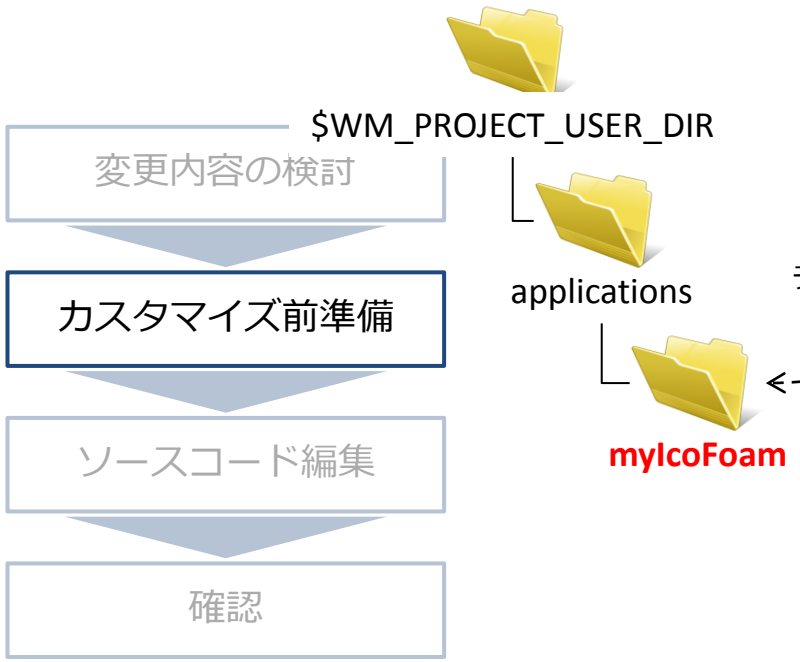
•ファイル編集：gedit <ファイルパス>

※.cppファイル(C++ソースファイル)：

インクルードファイルがc++専用の場合があり.cファイルのままだとビルドエラーの可能性

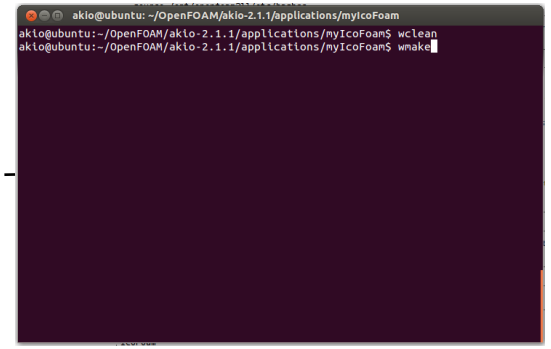
カスタマイズ前準備

④ビルドテスト



(1) カレントディレクトリを
カスタマイズソルバーディレクトリに

カレント
ディレクトリに



(2) 「wclean」を入力
（過去にビルドした中間ファイルなどを削除）

(3) 「wmake」を入力 (*)
（実行形式を作成）

(4) カスタマイズソルバーを実行してみる
⇒ベースソルバーと同じ結果が出力される(ハズ)

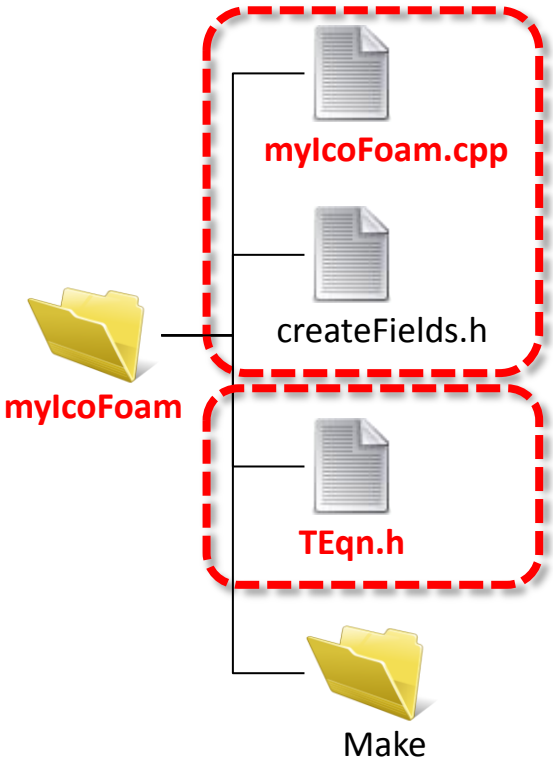
※ビルドエラーが発生した場合、下記確認
- ベースソルバーがビルド可能かどうか（ビルド不可能な場合、OpenFoam再インストールなど）
- 前ページのとおり変更しているかどうか

ソースコード編集

検討した解析モデルの導入

①スカラー場のデータ取り扱い指定
②スカラー場輸送方程式計算のコーディング
③ビルドエラー削除

- 変更内容の検討
- カスタマイズ前準備
- **ソースコード編集**
- 確認



変更対象ソースコード

追加ソースコード
(スカラー場輸送方程式計算)

ソースコード編集

①スカラー場のデータ取り扱い指定

createFields.h

既存ファイル変更

- 変更内容の検討
- カスタマイズ前準備
- ソースコード編集
- 確認

```
…
#include “createPhi.h”

Info<< "Reading field T¥n" << endl;
volScalarField T
(
    IOobject
    (
      "T",
      runTime.timeName(),
      mesh,
      IOobject::MUST_READ,
      IOobject::AUTO_WRITE
    ),
    mesh
);
….
```

計算開始時に
設定ファイルからスカラー場Tをロード
(myIcoFoam.cの計算開始時に
“#include “createFields.h”の記述あり)

スカラー場Tを設定ファイルから
ロードする命令を追加

ソースコード編集

②スカラー場輸送方程式計算のコーディング

TEqn.h

新規ファイル追加

myIcoFoam.cpp

既存ファイル変更

- 変更内容の検討
- カスタマイズ前準備
- **ソースコード編集**
- 確認

```
solve
(
    fvm::ddt(T)
  + fvm::div(phi, T)
  - fvm::laplacian(nu, T)
);
```

スカラー場Tを計算

$$\frac{\partial T}{\partial t}+\nabla\cdot(\mathbf{U}T)-\nabla\cdot(\nu\nabla T)=0$$

```
…
int main(int argc, char *argv[])
{
#include “setRootCase.h”
#include “createMesh.h”
#include “createFields.h”
#include “initCOntinuityErrs.h”
…
For(int corr=0; corr<nCorr; corr++)
{
#include “TEqn.h”
volScalarField rUA(1.0 / UEqn.A());
…
}
```

T計算のソースコードを
ロードする命令を追加

ソースコード編集

③ビルドエラー削除

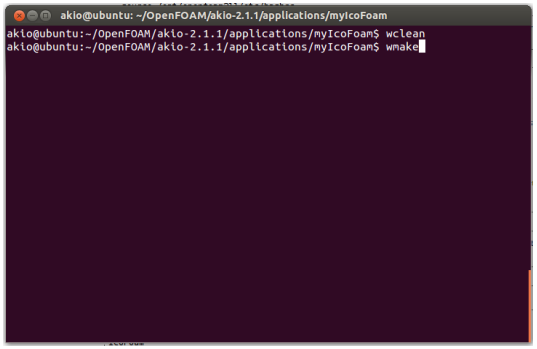
変更内容の検討

カスタマイズ前準備

ソースコード編集

確認

(1) カレントディレクトリを
カスタマイズソルバーディレクトリに



(2) 「wclean」を入力
（過去にビルドした中間ファイルなどを削除）

(3) 「wmake」を入力 (*)
（実行形式を作成）

エラーが発生した場合、ソースコード編集箇所を確認・修正

※ソースコード編集前にベースソルバーのビルドに成功しているので
ビルドエラー箇所は必ずソースコード編集箇所（or 編集漏れ）にある

確認

カスタマイズソルバーのテスト

①ケースファイルの作成

②カスタマイズソルバーの実行

③結果確認

確認

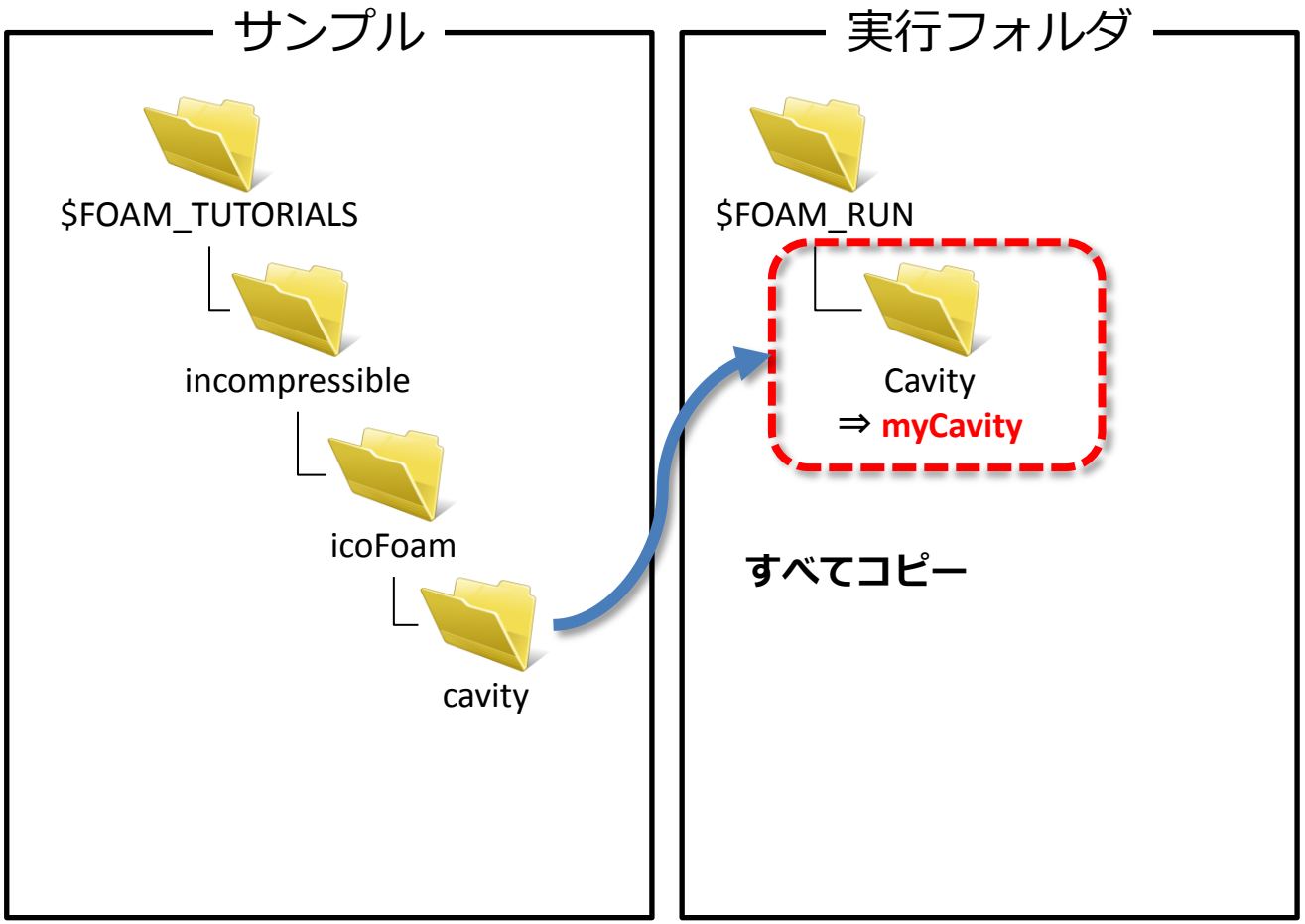
変更内容の検討

カスタマイズ前準備

ソースコード編集

確認

①ケースファイルの作成 – (1)



サンプルケースファイルから作成

確認

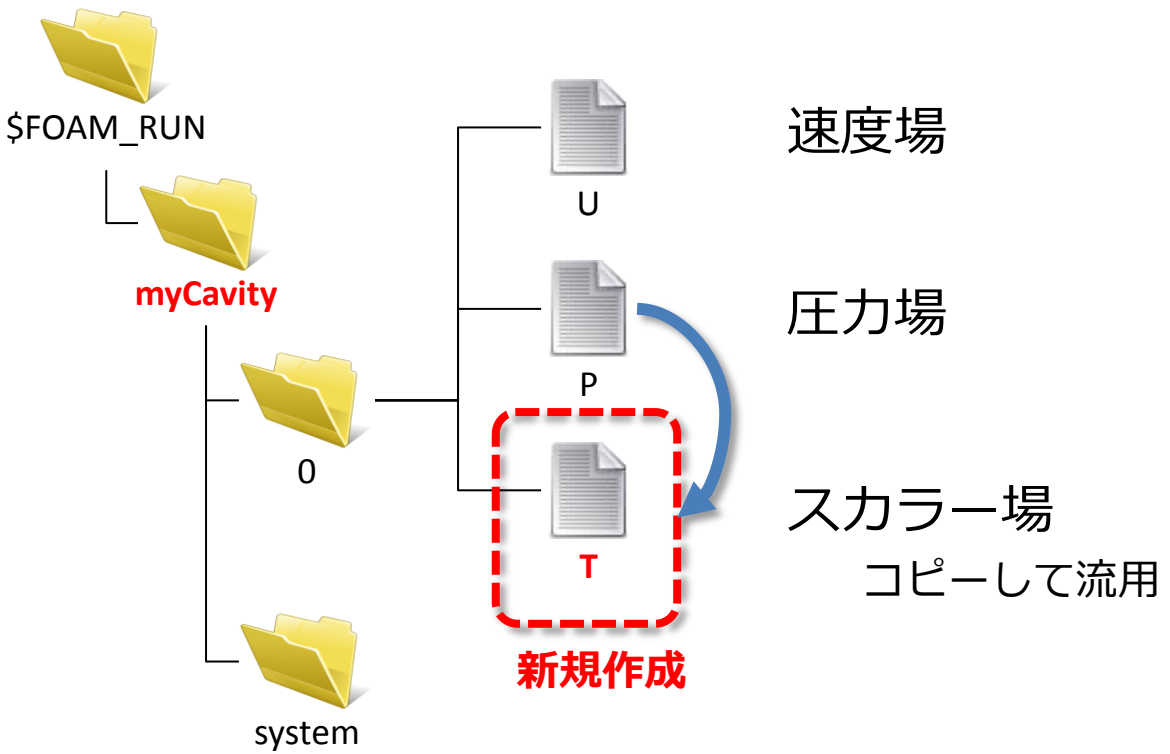
①ケースファイルの作成 – (2)

変更内容の検討

カスタマイズ前準備

ソースコード編集

確認



サンプルケースファイルから作成

確認

①ケースファイルの作成 – (3)

- 変更内容の検討
- カスタマイズ前準備
- ソースコード編集
- **確認**

新規ファイル追加（pからコピー）

T

```
dimensions [0 0 0 0 0 0 0];
internalField uniform 0;
boundaryField
{
    movingWall
    {
        type fixedValue;
        value uniform 1;
    }
    fixedWalls
    {
        type fixedValue;
        value uniform 0;
    }
    frontAndBack
    {
        type empty;
    }
}
```

既存ファイル変更

System/fvSchemes

```
…
divSchemes
{
    default none;
    div(phi,U) Gauss linear;
    div(phi,T) Gauss linear;
}
laplacianSchemes
{
    default none;
    laplacian(nu,U) Gauss linear corrected;
    laplacian((1|A(U)),p) Gauss linear corrected;
    laplacian(nu,T) Gauss linear corrected;
}
…
```

圧力場と同じ設定

確認

①ケースファイルの作成 – (4)

既存ファイル変更

変更内容の検討

カスタマイズ前準備

ソースコード編集

確認

system/fvSolution

```
...
solvers
{
    P
    {  ...
    }
    U
    {  ...
    }

    T
    {
        solver PBiCG;
        prconditioner DILU;
        tolerance 1e-05;
        relTol 0;
    }
}
...
```

圧力場と同じ設定

確認

②カスタマイズソルバーの実行

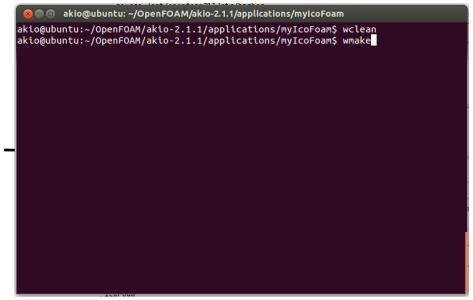
変更内容の検討

カスタマイズ前準備

ソースコード編集

確認

カレント
ディレクトリに



(1) カレントディレクトリを
ケースファイルのディレクトリに

(2) メッシュ作成
「blockMesh」

(3) カスタマイズソルバー実行
「**myIcoFoam**」

※シェルコマンド：

- 2つのファイルの違いを確認：diff <file1> <file2>
  違いがある行を出力

確認

③結果確認

変更内容の検討

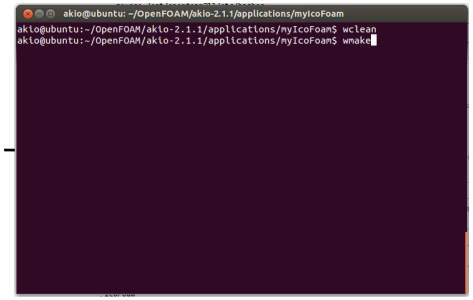
カスタマイズ前準備

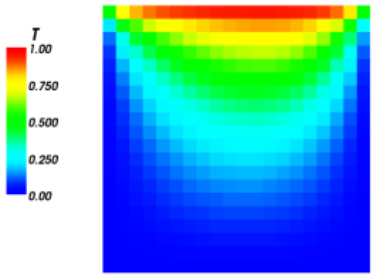
ソースコード編集

確認

(1) カレントディレクトリを
ケースファイルのディレクトリに
(前ページからそのまま)

カレント
ディレクトリに





オリジナルと
結果同じなはず

(2) 結果確認
- p, ph, Uの確認（diff / ParaFoam）
- Tの確認（結果の仮説 + ParaFoam）

※シェルコマンド：
- 2つのファイルの違いを確認：diff <file1> <file2>
  違いがある行を出力

まとめ

- 簡単なソルバーのカスタマイズ実施

参考文献

- PhD course in CFD
  with OpenSource software, 2009
  http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2009/

Appendix

実行環境

- Ubuntu 12.04.1 LTS
- OpenFoam v2.1.1