

C++, オブジェクト指向プログラミングの基礎

OpenFOAM[®]ソースコードの構造とコンパイル

2019年3月29, 30日
オープンCAE講習会@富山

富山県立大学 中川慎二

Disclaimer: OPENFOAM[®] is a registered trade mark of OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM[®] and OpenCFD[®] trade marks. This offering is not approved or endorsed by OpenCFD Limited.

注意

- OpenFOAMユーザーガイド, プログラマーズガイド, OpenFOAM Wiki, CFD Online, その他多くの情報を参考にしています。開発者, 情報発信者の皆様に深い謝意を表します。
- この講習内容は, 講師の個人的な経験 (主に, 卒 研究生等とのコードリーディング) から得た知識を共有するものです。この内容の正確性を保証することはできません。この情報を使用したことよって問題が生じた場合, その責任は負いかねますので, 予めご了承ください。

一部のクラス図の作成に, astah* community (無償版) を利用しています。

株式会社チェンジビジョン 社製 <http://astah.change-vision.com/ja/>

概要

- OpenFOAM の利用者を対象とし、OpenFOAMのソースコードの読み方の基本の基本を学びます。
- OpenFOAMのソースコードが、どのように整理されているのかを学びます。
- 実際にOpenFOAMのソースコードを見ながら、OpenFOAMのソースコードの特徴、ソースコード解読初心者が躓きやすい点などについても、解説します。

目次

- OpenFOAMとは？
- OpenFOAMとC++
 - クラス
 - ソルバ
 - ライブラリ
- OpenFOAMインストールとファイルの場所
- OpenFOAMソースコードの構造
 - ソルバ
 - ライブラリ
- OpenFOAMソースコードの調べ方

OpenFOAMの種類とバージョン

- OpenFOAM
 - オープンソースソフトウェア
 - GPLライセンス.誰もが自由に改変し,公開することができる。
- その結果,いくつかの枝分かれして,開発が進められている。
- 元は大学からスピンオフした企業で商用ソフトウェアとして販売されていた。その後,オープンソースソフトウェアとして公開された。開発元企業が,いくつかの企業に買収されるなど,いろいろな経緯があり,関連する企業・組織が複数存在する。

いろいろなバージョンについての,ほぼ完全なインストール方法解説

<https://openfoamwiki.net/index.php/Installation>

過去のバージョンの記録とインストール方法

<http://openfoam.org/download/archive/>

OpenFOAMの種類とバージョン

OpenFOAM Foundation

- 主たる開発 CFD Direct (OpenFOAM Foundationメンバーの一部)
- OpenFOAMの管理と公開 OpenFOAM Foundation
- <http://www.openfoam.org/>
- <https://cfd.direct/openfoam/about/>
- 最新リリース版 OpenFOAM v6

OpenCFD-ESI

- 主たる開発 OpenCFD
- OpenFOAMトレードマークの権利保持者 OpenCFD Ltd.
- OpenCFD Ltd. の所有者 ESI
- <https://www.openfoam.com/>
- <https://www.openfoam.com/history/>
- 最新リリース版 OpenFOAM v1812

各種バージョンのリリース日など

date	Foundation	OpenCFD-ESI	
2015/11/04	v 3.0.0		foam-extend-3.2
2016/06/28	v 4.0		
2016/06/30		v1606+ Foundation code 2016/04/21	foam-extend-4.0
2016/12/23		v1612+ Foundation code 2016/12/09	
2017/06/30		V1706 Foundation code 2017/04/26	
2017/07/25	v 5.0		
2017/12/31		v1712	
2018/06/29		v1806	
2018/07/10	v 6		
2018/12/20		v1812	

今回は, OpenFOAM v6 (Foundation版) を対象とします。

OpenFOAMインストール方法とファイルの格納場所

インストール方法

- Ubuntu Deb Pack
 - Ubuntu OS のパッケージ管理システムを使う。バージョンの組み合わせに制限あり。
 - 最も簡単・早い方法。コンパイル不要。
 - <https://openfoam.org/download/6-ubuntu/>
- Docker利用
 - コンテナ型仮想化Dockerを使う。いろいろなOSに対して、同じ環境を提供しやすい。
 - <https://openfoam.org/download/6-linux/>
- Source Pack
 - 汎用的方法。ソースコードをコンパイルする。
 - 手間がかかる。自由度は高い。
 - <https://openfoam.org/download/6-source/>
- Git Repository
 - 汎用的方法。バージョンコントロールシステム GIT を利用し、最新のソースコードを取得してコンパイルする。
 - リリース後のバグフィックスが受けられる。最新のバグを導入する可能性もある。

インストール方法

様々な環境への,様々なバージョンのインストール方法:Unofficial OpenFOAM wiki

<https://openfoamwiki.net/index.php/Installation>

インストール方法とファイルの場所

最新版 v6 の場合

- Ubuntu Deb Pack
 - 標準インストール先 `/opt/openfoam6`
 - 標準作業ディレクトリ `$HOME/OpenFOAM/user-6`
- Source Pack
 - 標準インストール先 `$HOME/OpenFOAM/OpenFOAM-6`
 - 標準作業ディレクトリ `$HOME/OpenFOAM/user-6`
- Git Repository (6のバグフィックス版)
 - 標準インストール先 `$HOME/OpenFOAM/OpenFOAM-6`
 - 標準作業ディレクトリ `$HOME/OpenFOAM/user-6`

そもそも、インストールとは？

- ファイルの配置
 - マシンが理解できる言葉で書かれた実行ファイル
- 実行に必要な情報のマシンへの通知
 - どこに、どのファイルがあるのか。
 - パスの設定
 - どのような条件・設定で動作するのか。
 - 環境変数の設定
 - OpenFOAMでは、「インストールディレクトリ /etc/bashrcファイル」を読み込むことで設定する
 - この設定を切り替えれば、複数バージョン併用可

OpenFOAMソースコードの構造

ソースコード ディレクトリ構造

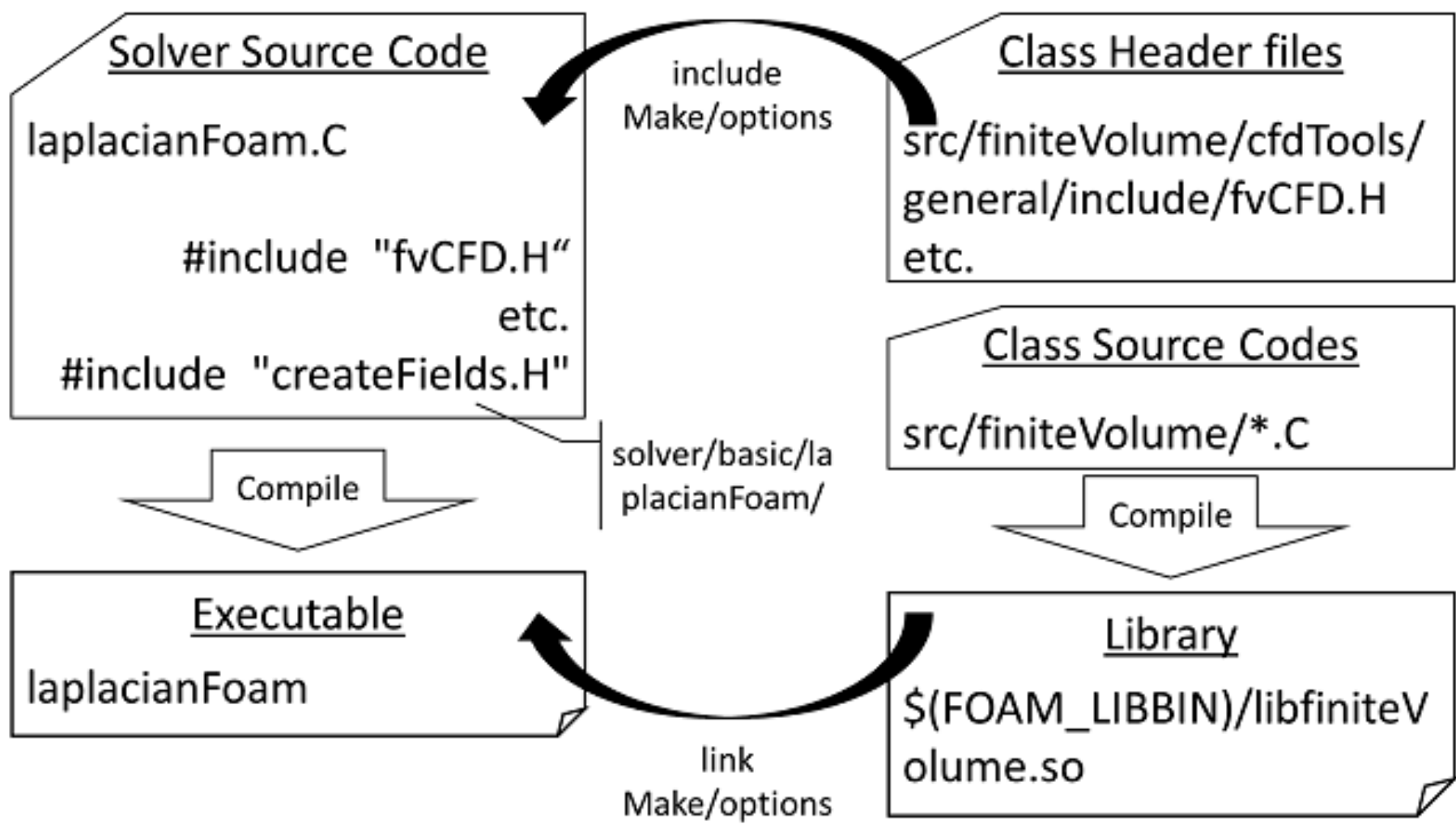
```
/opt/openfoam6$ tree -L 1
.
├── Allwmake
├── applications
├── bin
├── build-stamp
├── COPYING
├── doc
├── etc
├── platforms
├── README.org
├── README_scotch
├── src
├── test
├── tutorials
└── wmake
```

OpenFOAMソースコードの主要要素

- **src**: the core OpenFOAM source code
- **applications**: collections of library functionality wrapped up into applications, such as solvers and utilities
- **tutorials**: a suite of test cases that highlight a broad cross-section of OpenFOAM's capabilities
- **doc**: supporting documentation

ソルバのコンパイル

ライブラリのコンパイル

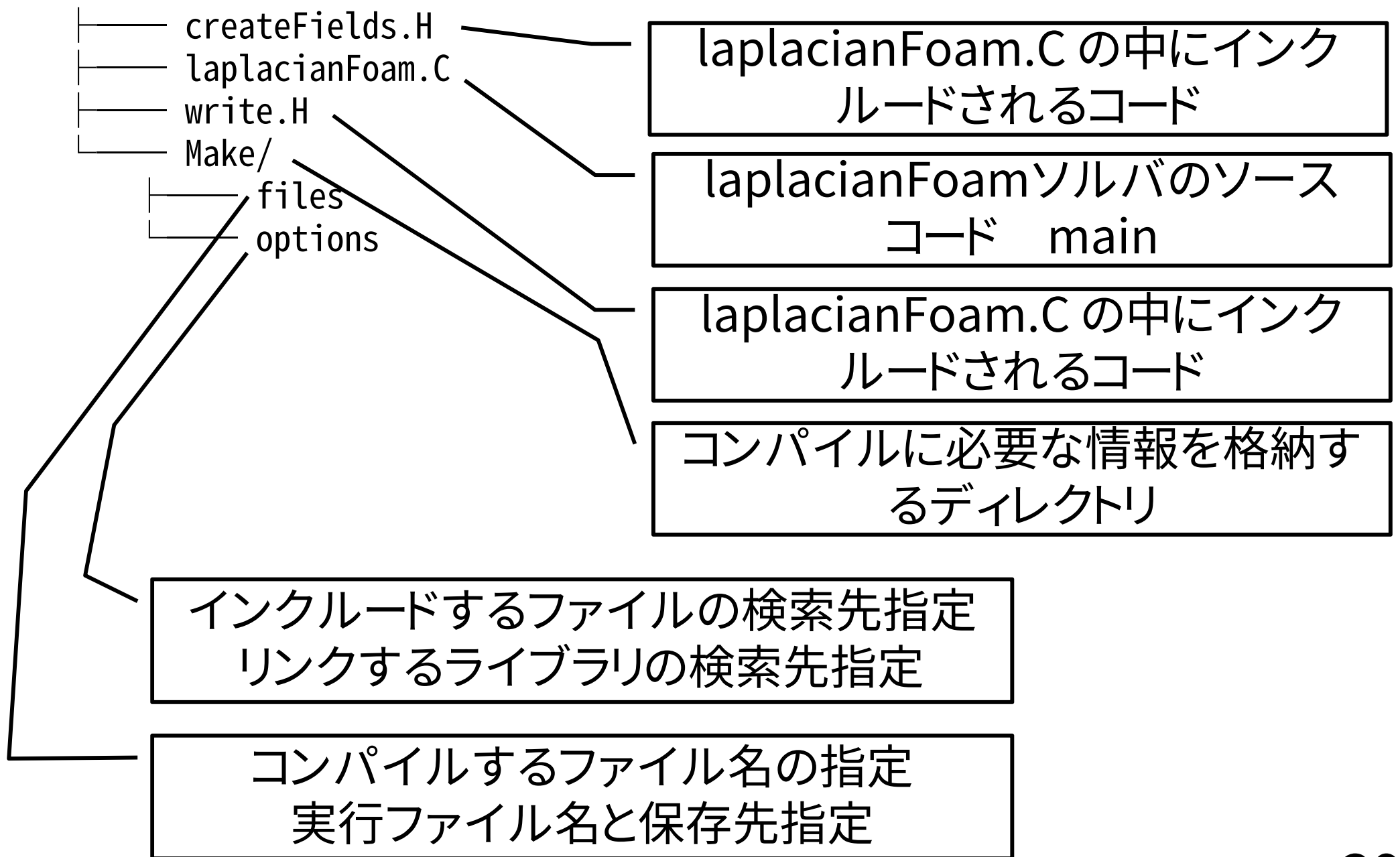


applications/solvers 例

```
/opt/openfoam6/applications/solvers$ tree -L 2
```

```
.
├── basic
│   ├── laplacianFoam
│   ├── potentialFoam
│   └── scalarTransportFoam
├── combustion
│   ├── chemFoam
│   ├── coldEngineFoam
│   ├── fireFoam
│   ├── PDRFoam
│   ├── reactingFoam
│   └── XiFoam
├── compressible
│   ├── rhoCentralFoam
│   ├── rhoPimpleFoam
│   ├── rhoSimpleFoam
│   └── sonicFoam
├── discreteMethods
│   ├── dsmc
│   └── molecularDynamics
├── DNS
│   └── dnsFoam
├── electromagnetics
│   ├── electrostaticFoam
│   ├── magneticFoam
│   └── mhdFoam
├── financial
│   └── financialFoam
├── heatTransfer
│   ├── buoyantBoussinesqPimpleFoam
│   ├── buoyantBoussinesqSimpleFoam
│   ├── buoyantPimpleFoam
│   ├── buoyantSimpleFoam
│   ├── chtMultiRegionFoam
│   └── thermoFoam
├── incompressible
│   ├── adjointShapeOptimizationFoam
│   ├── boundaryFoam
│   ├── icoFoam
│   ├── nonNewtonianIcoFoam
│   ├── pimpleFoam
│   ├── pisoFoam
│   ├── shallowWaterFoam
│   └── simpleFoam
├── lagrangian
│   ├── coalChemistryFoam
│   ├── DPMFoam
│   ├── icoUncoupledKinematicParcelFoam
│   ├── reactingParcelFoam
│   ├── sprayFoam
│   └── uncoupledKinematicParcelFoam
├── multiphase
│   ├── cavitatingFoam
│   ├── compressibleInterFoam
│   ├── compressibleMultiphaseInterFoam
│   ├── driftFluxFoam
│   ├── interFoam
│   ├── interPhaseChangeFoam
│   ├── multiphaseEulerFoam
│   ├── multiphaseInterFoam
│   ├── potentialFreeSurfaceFoam
│   ├── reactingEulerFoam
│   ├── twoLiquidMixingFoam
│   ├── twoPhaseEulerFoam
│   └── VoF
└── stressAnalysis
    ├── solidDisplacementFoam
    └── solidEquilibriumDisplacementFoam
```


applications/solvers/basic/laplacianFoam



src/transportModels 例

```
Allwmake*
├── compressible/
│   ├── compressibleTransportModel/
│   ├── lnInclude/
│   └── Make/
├── immiscibleIncompressibleTwoPhaseMixture/
│   ├── immiscibleIncompressibleTwoPhaseMixture.C
│   ├── immiscibleIncompressibleTwoPhaseMixture.H
│   ├── lnInclude/
│   └── Make/
├── incompressible/
│   ├── incompressibleTwoPhaseMixture/
│   ├── lnInclude/
│   ├── Make/
│   ├── singlePhaseTransportModel/
│   ├── transportModel/
│   └── viscosityModels/
├── interfaceProperties/
│   ├── interfaceCompression/
│   ├── interfaceProperties.C
│   ├── interfaceProperties.H
│   ├── lnInclude/
│   ├── Make/
│   └── surfaceTensionModels
├── twoPhaseMixture/
│   ├── lnInclude/
│   ├── Make/
│   └── twoPhaseMixture/
└── twoPhaseProperties/
    ├── alphaContactAngle/
    ├── alphaFixedPressure/
    ├── lnInclude/
    └── Make/
```

Makeディレクトリが存在
コンパイルする単位:ライブラリ

incompressibleTransportModels
ライブラリのディレクトリ(Make/files参照)

viscosityMedelクラスと,その派
生クラスが格納される

src/transportModels/ incompressible/ viscosityModels

- BirdCarreau
 - BirdCarreau.C
 - BirdCarreau.H
- Casson
 - Casson.C
 - Casson.H
- CrossPowerLaw
 - CrossPowerLaw.C
 - CrossPowerLaw.H
- HerschelBulkley
 - HerschelBulkley.C
 - HerschelBulkley.H
- Newtonian
 - Newtonian.C
 - Newtonian.H
- powerLaw
 - powerLaw.C
 - powerLaw.H
- strainRateFunction
 - strainRateFunction.C
 - strainRateFunction.H
- viscosityModel
 - viscosityModel.C
 - viscosityModel.H
 - viscosityModelNew.C

namespace
Foam::viscosityModels::BirdCarreau
継承
class BirdCarreau: public viscosityModel

BirdCarreauクラスの定義

BirdCarreauクラスの宣言

Class
Foam::viscosityModel
Description
An abstract base class for
incompressible viscosityModels.

OpenFOAMのコンパイル laplacianFoamを例に

OpenFOAMのコンパイル

- wmakeコマンド
 - /opt/openfoam6/wmake/wmake
 - 必要な様々な設定・作業を実施のうえでmake

Description

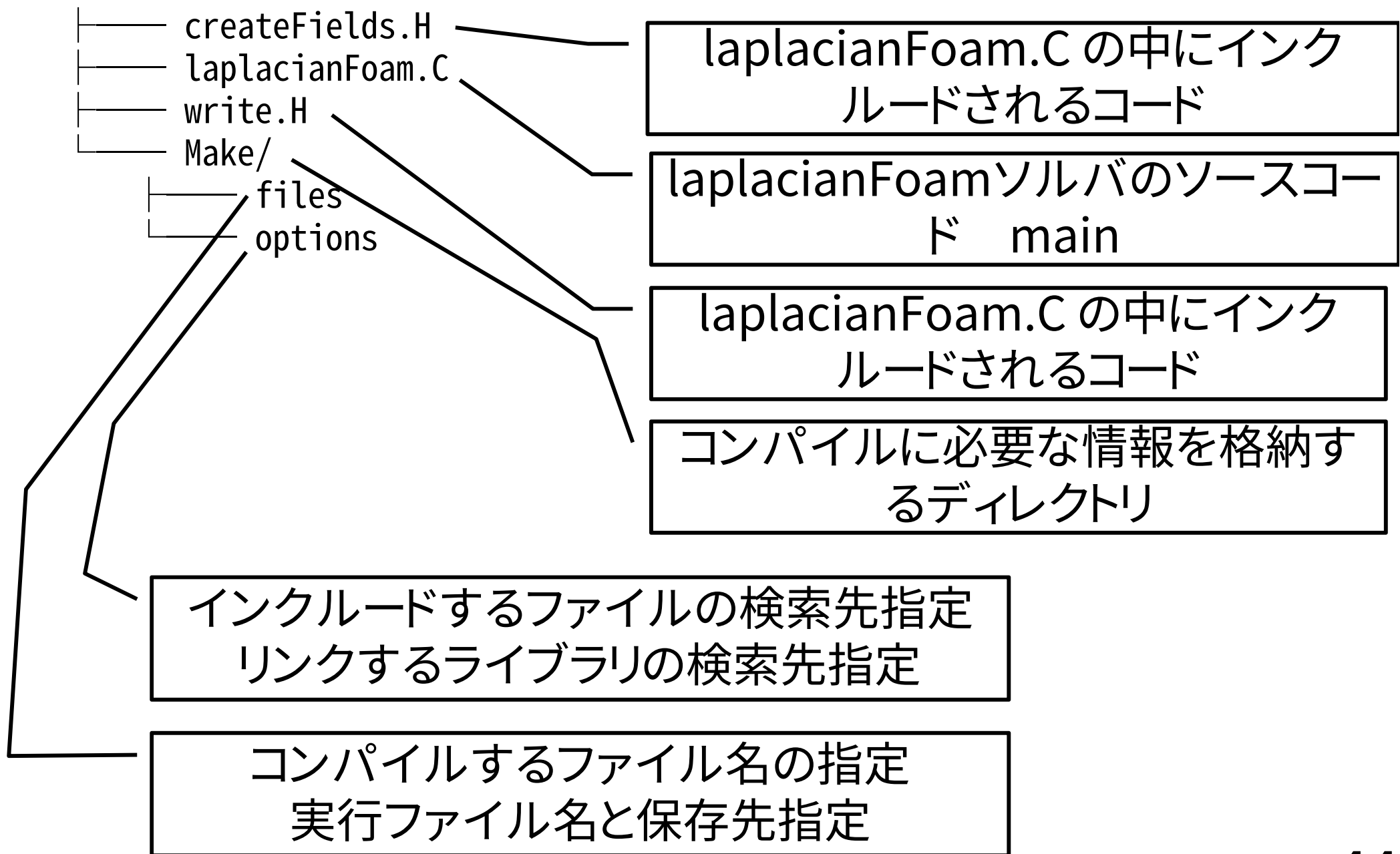
General, easy to use make system for multi-platform development

with support for local and network parallel compilation.

共通設定

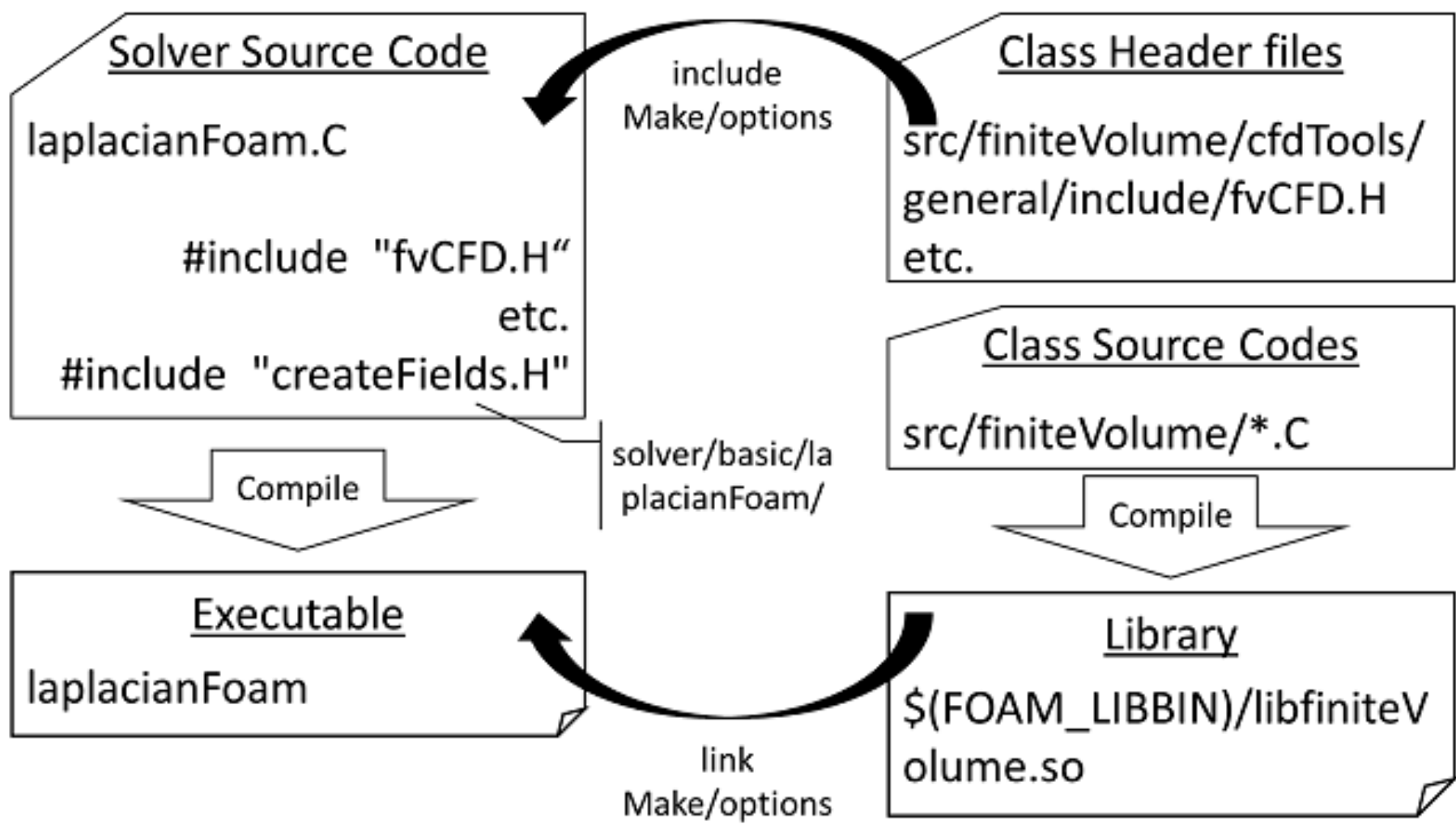
- /opt/openfoam6/wmake/makefiles/

applications/solvers/basic/laplacianFoam

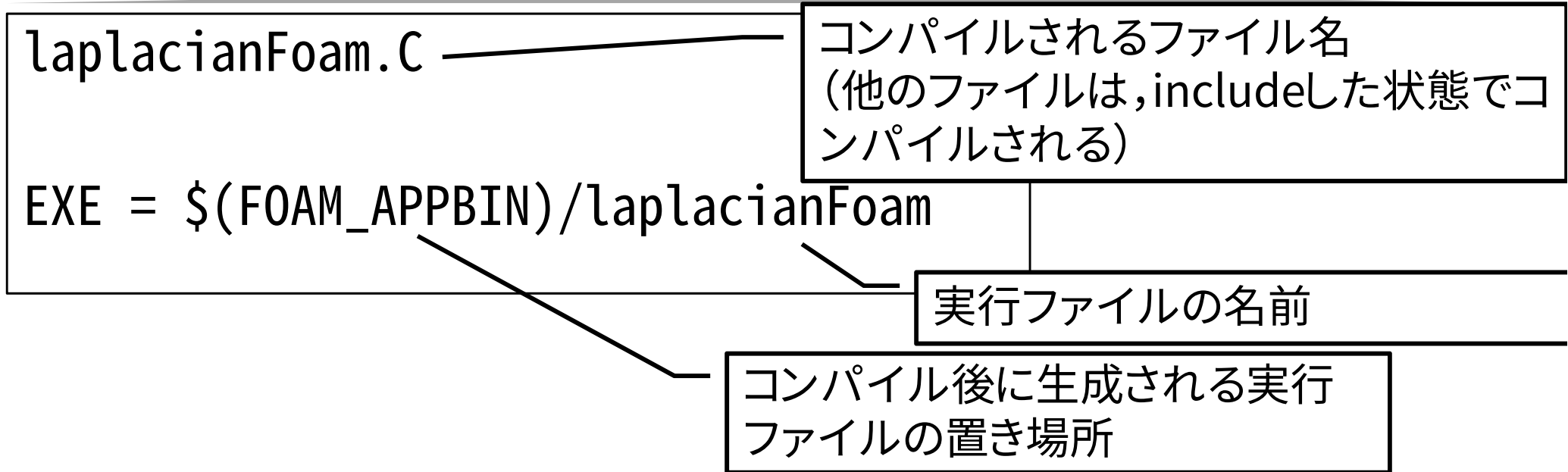


ソルバのコンパイル

ライブラリのコンパイル



laplacianFoam/Make/files



- 先頭の\$は変数であることを示す
 - FOAM_APPBIN 標準の実行ファイル置き場
 - FOAM_USER_APPBIN ユーザ固有の実行ファイル置き場
- 端末から echo \$FOAM_APPBINで内容確認

laplacianFoam/Make/options

EXE_INC = ¥

-I\$(LIB_SRC)/finiteVolume/lnInclude ¥

-I\$(LIB_SRC)/meshTools/lnInclude

インクルードするファイルの検索先指定

行末の記号は、行の継続を示すバックスラッシュ(/)

EXE_LIBS = ¥

-lfiniteVolume ¥

-lfvOptions ¥

-lmeshTools

リンクするライブラリの検索先指定

- LIB_SRC = \$(WM_PROJECT_DIR)/src

- 次の順序で、ヘッダファイルが検索される。
 - ソルバ・ソースコードディレクトリ内にある `lnInclude` ディレクトリ
 - ソルバ・ソースコードディレクトリ
 - OpenFOAMで規定した `PROJECT_INC`
 - `Make/options`ファイル内で `-I` オプションによって指定したディレクトリ。
[`$(LIB_SRC)/finiteVolume/lnInclude`]

- wmake/makefiles/general でPROJECT_INC
として

```
PROJECT_INC = -I$(LIB_SRC)/$(WM_PROJECT)/lnInclude ¥  
              -I$(LIB_SRC)/OSspecific/$(WM_OSTYPE)/lnInclude
```

- 具体的には

```
$(WM_PROJECT_DIR)/src/OpenFOAM/lnInclude  
$(WM_PROJECT_DIR)/src/OSspecific/POSIX/lnInclude
```

ライブラリへのリンク

- ソルバ自身に含まれないクラスは, 独立してコンパイルし, 実行形式のバイナリファイルを作成する。
- このファイルは, shared object library であり, 拡張子 `.so` がつく。
- ソルバは, このように独立してコンパイル済みの `.so` ファイルを実行時に呼び出すことができる。dynamic link と呼ぶ方式である。
- `laplacianFoam` では, `EXE_LIBS = -lfiniteVolume` と指定している。この時, `$(FOAM_LIBBIN)/libfiniteVolume.so` が実行時に利用される。

ライブラリへのリンク

- optionsファイルでのライブラリ指定時には、ファイル名先頭のlibと、拡張子.soとを除外した名前を使用する。つまり、-lfiniteVolumeという記述があれば、実際に使われるライブラリのファイル名は、libfiniteVolume.so である。
- この libfiniteVolume.so は、src/finiteVolume に存在するソースコードから生成されたものである。src/finiteVolume/Make/filesファイルを読むと、それが確認できる。
- srcディレクトリ以下には多くのディレクトリが存在する。Makeディレクトリを含むディレクトリでは、ライブラリが作成されている。

ライブラリの検索場所

次の場所から,リンクする shared object library を見つける。

- \$FOAM_LIBBIN ディレクトリ
- 環境に依存するディレクトリ
\$WM_DIR/rules/\$WM_ARCH/ directory
- Make/optionsファイル内で -L オプションによって指定したディレクトリ

- Make/optionsファイル内で -l オプションによって指定したライブラリ。
- ただし,ここで指定する名前は,ライブラリファイル名の先頭にあるlibと拡張子(.so)を取り除いたものである。
- 例えば,-lfiniteVolume と指定すると,libfiniteVolume.so が実際のファイルとなる。

- wmake/makefiles/general で
PROJECT_LIBS = -lOpenFOAM と設定している。

OpenFOAMソースコードの調べ方

調べ方

- OpenFOAM C++ Source Guide
 - コード自体から, Doxygen を使って生成
 - www.openfoam.org/docs/cpp
 - (注意) 上記サイトは最新リリース版
 - 旧版もあり <https://cpp.openfoam.org/v5/> など
 - クラス説明, コード説明, 継承関係図など
 - ローカルマシン上でも, Doxygen を使って, 同じ情報を作成できる
 - 注意: 大量のファイルが生成され, 2GB以上の容量が必要。長時間の作業となる。
 - `sudo apt-get install doxygen graphviz`
 - `cd $WM_PROJECT_DIR/doc/Doxygen`
 - `./Allwmake`

OpenFOAM C++ SourceGuide 注意

- Class Reference と File Reference では, 表示される内容が異なる
- クラスの全容を知りたい時には, Class Reference を参照するとよい。

OpenFOAM v6
The OpenFOAM Foundation

C++ Source Code Guide

Main Page Namespaces Classes Files Search

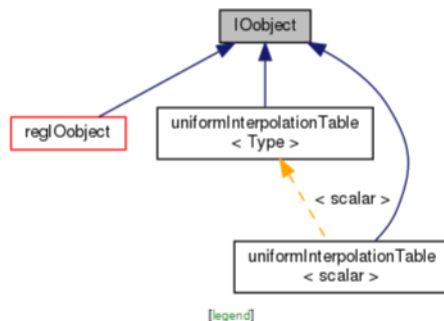
Foam > IObject >

IObject Class Reference

Public Types | Public Member Functions | Static Public Member Functions | Static Public Attributes | Protected Member Functions | List of all members

IObject defines the attributes of an object for which implicit **objectRegistry** management is supported, and provides the infrastructure for performing stream I/O. [More...](#)

Inheritance diagram for IObject:



Collaboration diagram for IObject:

NamedEnum< fileCheckTypes, 4 >

OpenFOAM v6
The OpenFOAM Foundation

C++ Source Code Guide

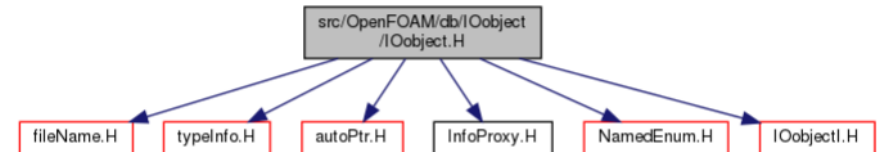
Main Page Namespaces Classes Files Search

src > OpenFOAM > db > IObject >

IObject.H File Reference

Classes | Namespaces | Functions

Include dependency graph for IObject.H:



This graph shows which files directly or indirectly include this file:



ソースコードを読み解くために

- 変数のタイプ(クラス)を意識
 - volScalarField, dimensionedScalar ? など
- まずは, 宣言(*.H)を見て, 流れをつかむ。
 - 関数は, 何(クラス)を受け取り, 何を返すか?
- Slow and steady wins the race
 - 少しずつ, 理解を深める
 - 小さな部分の積み重ね
 - 繰り返す, 繰り返す, 繰り返す
- 基礎を学習
 - 現象・式とソースの両方を学ぶ
- 一般的なデザインパターンの理解を深める

よく使う クラス

	非フィールド値 場所によらず一定	フィールド値 場所によって値が変わる	
		セル面での値	セル体積(中心)での値
スカラー	<code>dimensionedScalar</code> <code>dimensioned<scalar></code>	<code>surfaceScalarField</code> <code>GeometricField<scalar,</code> <code>fvPatchField, surfaceMesh></code>	<code>volScalarField</code> <code>GeometricField<scalar,</code> <code>fvPatchField, volMesh></code>
ベクトル	<code>dimensionedVector</code> <code>dimensioned<vector></code>	<code>surfaceVectorField</code> <code>GeometricField<vector,</code> <code>fvPatchField, surfaceMesh></code>	<code>volVectorField</code> <code>GeometricField<vector,</code> <code>fvPatchField, volMesh></code>

上の行は, typedefで定義された別名
下の行が本来の定義

templateクラスで多様なタイプに対応

TとDTの定義:createFields.H

volScalarField T

volScalarField T(IOobject, mesh);
Tという名前で、volScalarFieldクラスのオブジェクトを作成する。引数を2つ渡してコンストラクタ指定。

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

dimensionedScalar DT(Istream& is);
DTという名前の dimensionedScalarクラスのオブジェクトを作成する。引数はIstream&。読み取った値で初期化する。

dimensionedScalar DT

```
(  
  transportProperties.lookup("DT")  
);
```

transportPropertiesオブジェクトのlookup関数を使って、transportPropertiesディクショナリのDTという項目の値を読み込む。transportPropertiesは、IOdictionaryクラスのオブジェクト。

IOdictionary transportProperties
lookup関数の戻り値は、Istream&

調べ方

- デバッガを使用する。
 - HowTo debugging
http://openfoamwiki.net/index.php/HowTo_debugging
- debugSwitchesを利用して,実行時に追加メッセージを表示させる。
- Linuxのfindコマンドを使って,ファイルを探す。

参考資料

- OpenFOAM ProgrammersGuide, UserGuide
- ソースコード
- Imperial College 博士論文など
 - Hrvoje Jasak, Henrik Rusche, Franjo Juretic などなど
 - <http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/docs/>
- PENGUINITIS!
 - <http://penguinitis.g1.xrea.com/study/OpenFOAM/index.html>
- <http://openfoamwiki.net/>
- <http://www.cfd-online.com>

-
- OpenFOAM のための 便利なLinuxコマンド解説
– <http://cfd.direct/openfoam/linux-guide/>

Try: icoFoamソルバ名称変更

```
mkdir -p $WWM_PROJECT_USER_DIR/solvers
```

```
cp -r  
$WWM_PROJECT_DIR/applications/solvers/incompressible/icoFoam  
$WWM_PROJECT_USER_DIR/solvers/myIcoFoam/
```

1行に続けて書いてあります。

```
cd $WWM_PROJECT_USER_DIR/solvers/myIcoFoam/
```

```
mv icoFoam.C myIcoFoam.C
```

```
sed -i -e 's/icoFoam/myIcoFoam/g' ./Make/files
```

```
sed -i -e 's/FOAM_APPBIN/FOAM_USER_APPBIN/g' ./Make/files
```

```
wmake
```

Try: myIcoFoam 例題実行

```
cp -r  
$FOAM_TUTORIALS/incompressible/icoFoam/c  
avity/cavity/ $FOAM_RUN/cavity
```

```
run
```

```
cd cavity
```

```
blockMesh
```

```
myIcoFoam
```