PENNSTATE
6th OpenFOAM® Workshop
PennState University, USA
13-16 June 2011

Integrated Development Environment (IDE) Eclipse for OpenFOAM®

Assessing the Perfomance of bubbleFoam

Astrid Mahrla*†[1] and Holger Marschall[2]

[1]Chair of Chemical Engineering, Technische Universität München
[2]Center of Smart Interfaces, Technische Universität Darmstadt

June 13, 2011

1 Introduction

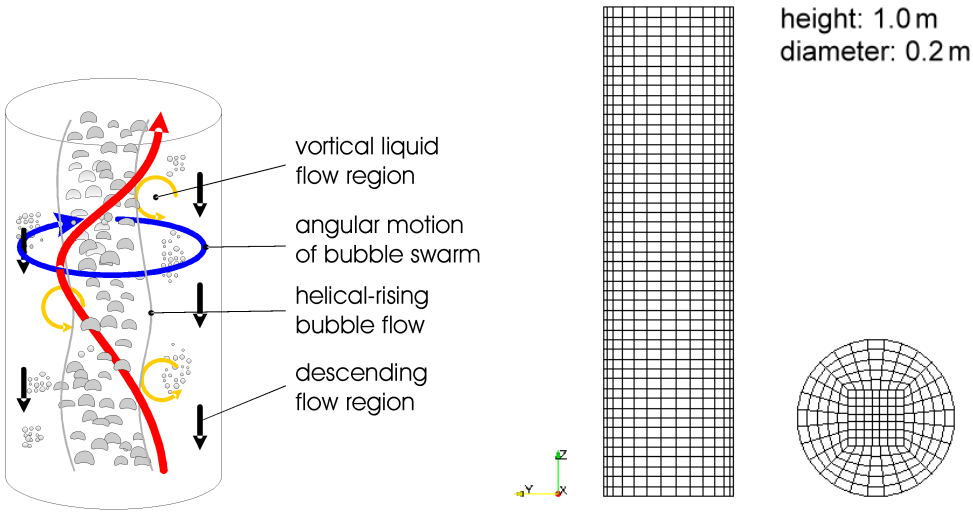
"Eclipse is an open source community whose projects are focused on building an extensible development platform, runtimes and application frameworks for building, deploying and managing software across the entire software lifecycle. Many people know us, and hopefully love us, as a Java IDE but Eclipse is much more than a Java IDE." - www.eclipse.org

Scope and objective of this tutorial is the introduction of the *Integrated Development Environment* (IDE) `Eclipse` for OpenFOAM®. `Eclipse` is a powerful IDE originally developed for Java programming. But with the *C/C++ Development Toolkit* (CDT) extension Eclipse becomes a very common IDE for *fast and efficient* C++ programming.

Due to the amount of advantages of using an IDE only a few of them are listed below.

- Well-arranged graphical user interface offering project explorer, outline, ...
- Fully integrated powerful text editor offering code highlighting, autocompletion, ...
- Integrated compiler offering linked error and warning marks
- Integrated debugger and debugging environment offering breakpoints and variable information
- Project management: bookmarks and tasks
- Extension: version management, multiple language support (Java, Python, ...)

†Corresponding Author: Astrid Mahrla (Astrid.Mahrla@ch.tum.de)



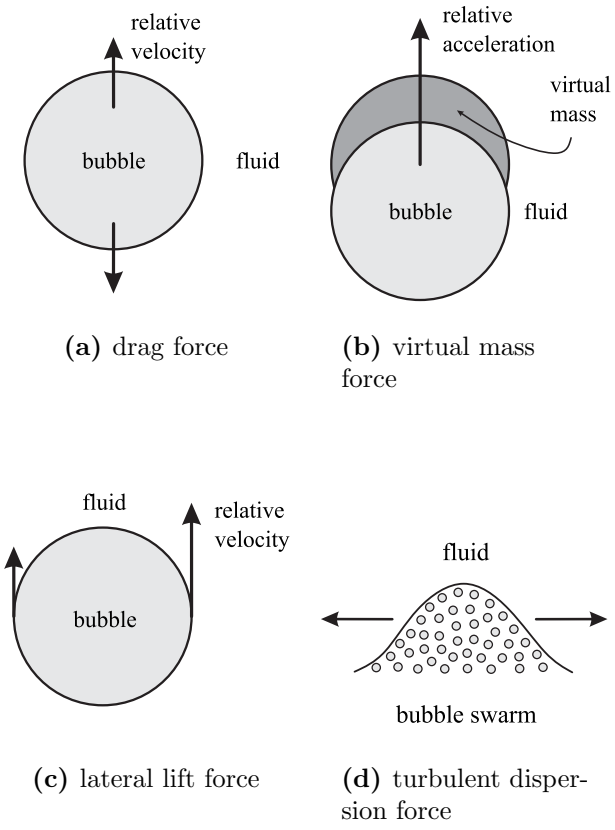
(a) bubbly flow phenomena **(b)** bubble column mesh

Figure 1: Bubbly flow in a bubble column

For further information concerning `Eclipse` refer to [1, 2, 3].

As OpenFOAM deals with physics we shall do so as well for the introduction of Eclipse: in this tutorial we will deal with bubbly flows (as they occur in bubble columns). In OpenFOAM the `bubbleFoam` solver enables to simulate the complex flow dynamics of these kind of gas-liquid flows. Currently, there is only the skeleton of a state-of-the-art model available and implemented in `bubbleFoam`: a two-fluid model framework based upon the Eulerian-Eulerian method. In this model, the flow morphology (i.e., the bubbles' shape) is not resolved explicitly at all, but is taken into account in an averaged manner presuming a specific shape. I.e., *conditional volume-averaging* of two-phase conservation equations results in the concept of interpenetrating continua [5], in which *all phase interactions* have to be modeled in order to physically close the system of governing equations. Thus, closure modeling is of major importance in order to gather reliable results. We will have a look at bubbly forces, which can be categorized further into drag and non-drag forces (→ slides). These forces essentially characterize the fluid dynamics of the two-phase flow system present in a bubble column.

The **motivation** of this tutorial rests in the fact that the already existing `bubbleFoam` solver needs to be restructured: i.e., we want to implement `runTime`-selective models for the calculation of interfacial forces. For this purpose, the existing 'hard-coded' models have to be rearranged in `C++` libraries. Moreover, new models for both drag and non-drag forces have to be added. The rearrangement to libraries allows runTime-selective access to the interfacial force models during the simulation. The remainder of this tutorial will explain how this can be accomplished using `Eclipse`.



(a) drag force

(b) virtual mass force

(c) lateral lift force

(d) turbulent dispersion force

Figure 2: Bubble Forces – drag and non-drag forces [4]

2 Adding a runTime-selective Model – Drag Force

2.1 Preliminary steps

Start the OpenFOAM-1.6-ext-dbg terminal, as the debugging in the following tutorial will need the debug compiled binaries. Please export the path to the GNU Debugger (GDB) for Eclipse.

- `export PATH=/home/ubuntu/gdb4Eclipse/bin:$PATH`

Begin creating a personal version of `bubbleFoam` solver and copy the `cylindricBubbleColumn` test case in the correct `$WM_PROJECT_USER_DIR` directories.

- `bubbleFoam` solver:
  `/usr/lib/OpenFOAM-1.6-ext-dbg/applications/solvers/multiphase/bubbleFoam`
- `cylindricBubbleColumn` test case:
  `/cdrom/OFW6/Training/case-cyclindricBubbleColumn.tgz`.

Change the name of the `bubbleFoam` folder into `bubbleFoamMod` and run `blockMesh` on the `cylindricBubbleColumn`.

2.2 Modifying bubbleFoam

1. **Setting up Eclipse for OpenFOAM**
   (a) Launch `Eclipse` using `eclipse &` and choose your workspace. If you're developing several projects it's advisable to create a workspace folders for each project. Herein we use the default workspace.
   (b) Change the developing environment to `C++` in the menu bar under *Window* → *Show View* → *Other..* → *C/C++ Projects*.
   (c) Make sure that *Project* → *Automatically build* in the menu bar is *un*checked.
   (d) Import the `bubbleFoamMod` solver by creating a new `C++` project in the menu bar under *Files* → *New* → *C++ Project*. Deactivate *Use default location*, then select the `bubbleFoamMod` folder. Set `bubbleFoamMod` as name for the project and click *Finish* (see Figure 3).
   (e) For using the `OpenFOAM`-specific compilation script `wmake`, change the compiler properties for `Eclipse`. Right-click on your new project in the project explorer on the left side, select *properties*. Setting the build command under *C/C++ Build*, deactivate the default build command and choose `OpenFOAM`'s `wmake`. Deactivate *Generate Makefiles automatically*. Set the build directory - maybe you have to remove the `/Release` or `/Debug`. Confirm with *OK* (see Figure 4).

2. **Developing and compiling with Eclipse**
   (a) For the actual developing and compiling procedure, open your project folder in the project explorer window and double click a file, so the editor will open the file in the middle of your screen. You can now edit your file comfortably with the `Eclipse` text editor. Make line number visible by *Right-Click* → *Preferences..* → *Editor*. Click on *Text Editors* and check *Show line numbers*.

(b) For adapting your new solver `bubbleFoamMod` make the adequate changes in `Make/files` and rename `bubbleFoam.C` to `bubbleFoamMod.C` (see Figure 5). Moreover, place the executable at `EXE = $(FOAM_USER_APPBIN)/bubbleFoamMod`.

(c) Recompile the solver by adding the new make targets `wmake` and `wclean`.

- First select the *Make Targets*-tab and select the directory where your `Make` folder is located. Create a new make target, by clicking on the *new make target* button. For compiling an application give the target a descriptive name – we choose `wmake` - leave the make target plain. The default build command is `wmake`.
- Create the corresponding `wclean` – do not forget to change the build command to `wclean`.
- Execute the *make targets* by double-clicking.

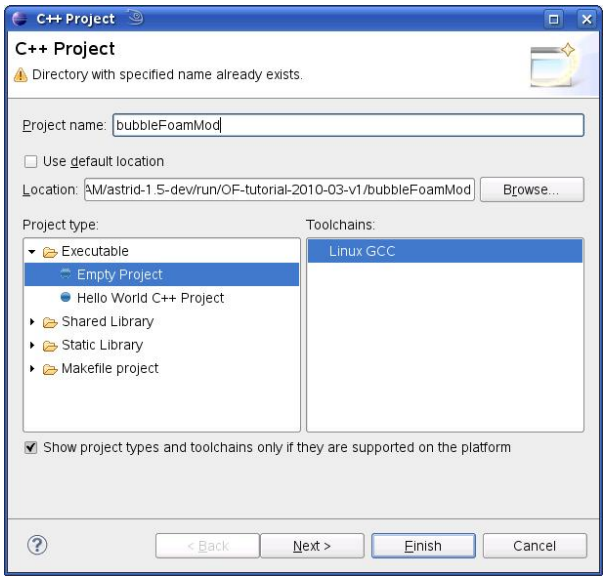


Figure 3: Import of bubbleFoamMod

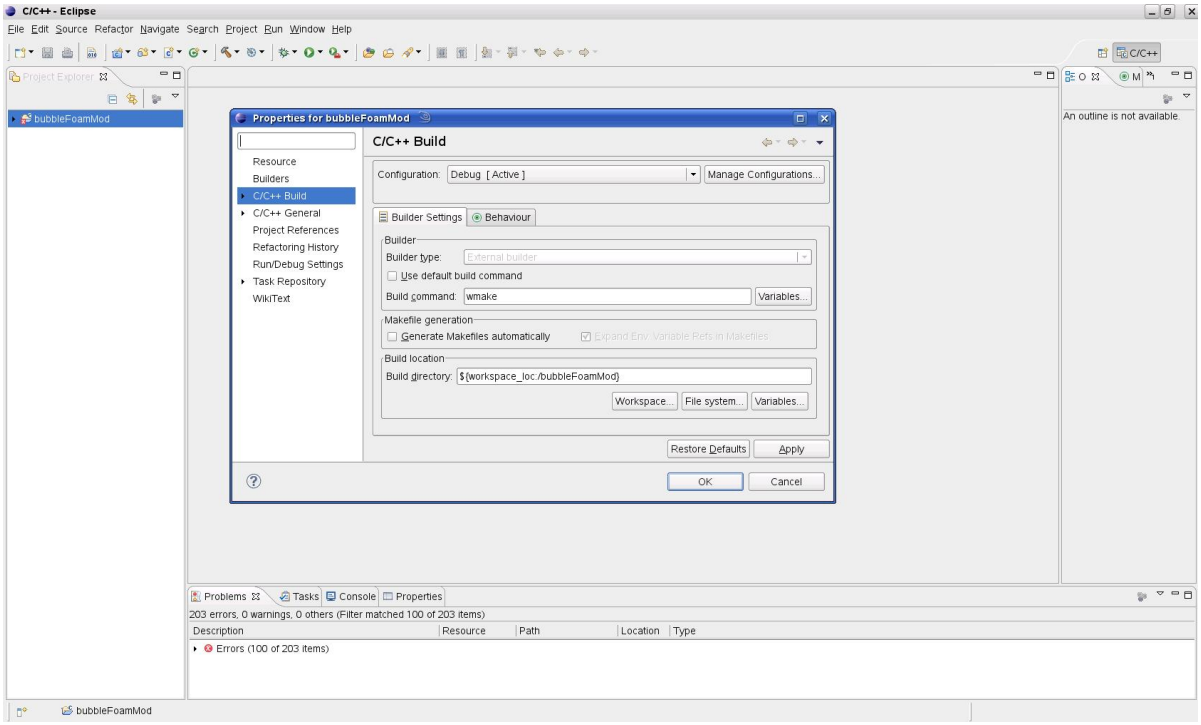


Figure 4: Eclipse Compiler Settings

```
dragModels/dragModel/dragModel.C
dragModels/dragModel/newDragModel.C
dragModels/SchillerNaumann/SchillerNaumann.C

LIB = $(FOAM_USER_LIBBIN)/libEulerianInterfacialModels
```

Source Code 1: interfacialModels/Make/files

```
EXE_INC = \
    -I$(LIB_SRC)/finiteVolume/lnInclude \
    -I../phaseModel/lnInclude

LIB_LIBS = \
    -L$(FOAM_USER_LIBBIN) -lphaseModel
```

Source Code 2: interfacialModels/Make/options

```
phaseModel/phaseModel.C

LIB = $(FOAM_USER_LIBBIN)/libphaseModel
```

Source Code 3: phaseModel/Make/files

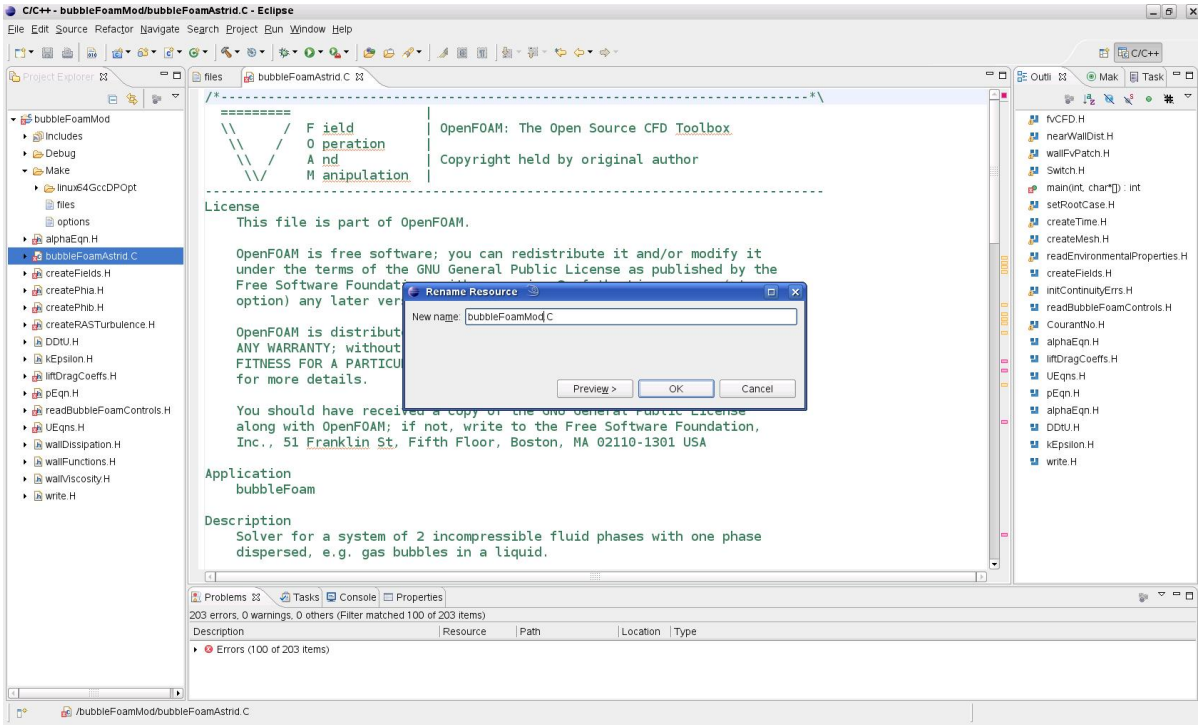


Figure 5: Eclipse Working Environment - Renaming bubbleFoamMod

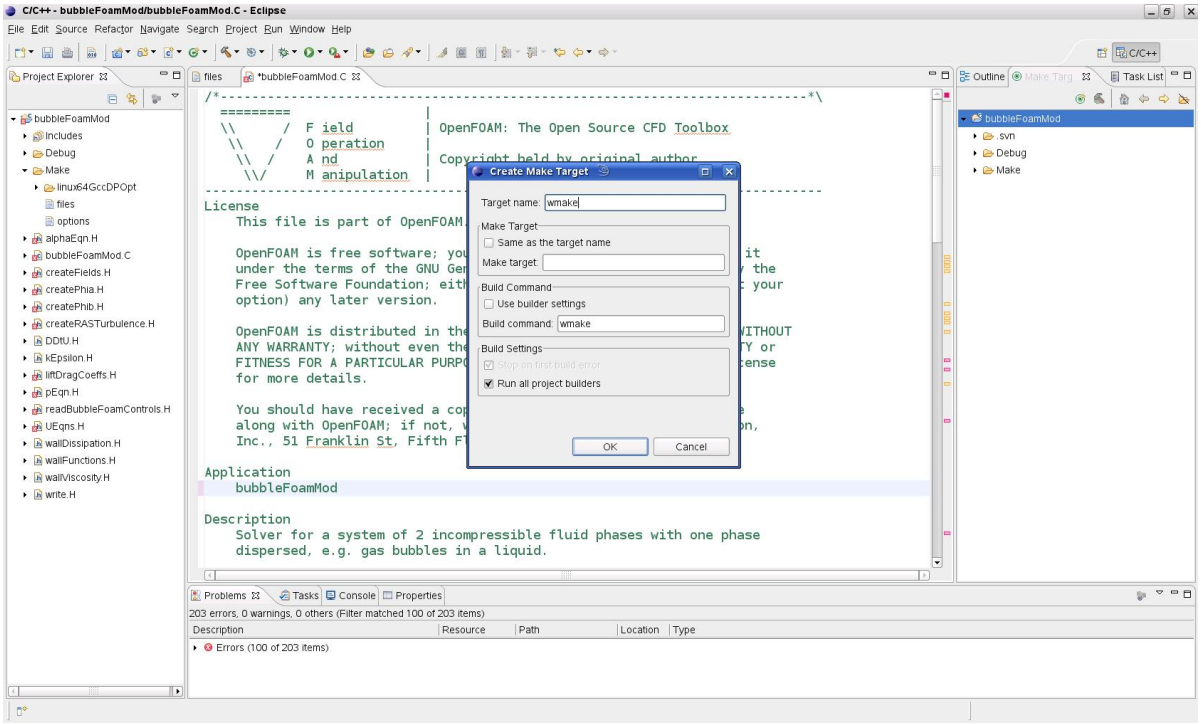


Figure 6: Create make targets

```
EXE_INC = \
    -I$(LIB_SRC)/finiteVolume/lnInclude \
    -I$(LIB_SRC)/transportModels/incompressible/lnInclude \
    -IinterfacialModels/lnInclude \
    -IphaseModel/lnInclude \
    -Iaveraging

EXE_LIBS = \
    -lfiniteVolume \
    -lmeshTools \
    -lincompressibleTransportModels \
    -L$(FOAM_USER_LIBBIN) -lphaseModel \
    -L$(FOAM_USER_LIBBIN) -lEulerianInterfacialModels
```

Source Code 4: Make/options

3. **Altering bubbleFoam**

After the change of the name and recompiling the new solver, we will implement the drag and phase models as their own classes instead of the hard-coded version as currently implemented. Therefore, take some files from the `twoPhaseEulerFoam` solver, where some physical models for fluidized beds are already implemented dynamically.

- `twoPhaseEulerFoam` solver:
  `/usr/lib/OpenFOAM-1.6-ext-dbg/applications/solvers/multiphase/..../twoPhaseEulerFoam` with subfolders `interfacialModels` and `phaseModel`

(a) Return to the terminal, copy the folders `interfacialModels/` and `phaseModel/` into the main `bubbleFoamMod` directory.

(b) Return to `Eclipse` and refresh your workspace under *File → Refresh (F5)*.

(c) Delete all subdirectories in `interfacialModels/dragModels` except `dragModel/` and `SchillerNaumann/`.

(d) In the `Make`-folder of `interfacialModels/` adapt the files `files` (deleting the unused drag models) and `options`. The files should look now as given in the Sources 1 and 2.

(e) Change `phaseModel/Make/files` appropriately. Eventually the file should read as given in Source 3.

(f) Next in the `Make`-directory of bubbleFoam, include the new models in the file `options`. The file should look as illustrated in Source 4.

(g) For having access to the new classes, phase model pointers for each phase a and b have to be defined and initialized in `createFields.H` (Source 5). The models will be read in as an entry in the `transportProperties` dictionary of the case. Use the arrow operator –> in order to call the functions for the phase properties as density, viscosity and diameter.

(h) Next a new dictionary has to be created in order to make runTime-selectivity available for all interfacial force models we are going to add. Therefore define the new dictionary `interfacialProperties` in `createFields.H` (Source 6) and implement one drag model pointer for each phase a and b.

(i) Include the `dragModel.H` file in `bubbleFoamMod.C` as shown in Source 7, so it is available for the compiler and then for our new runTime-selective application.

(j) Finally, alter the `liftDragCoeffs.H` by inserting the correct calculation of the drag coefficient using the new drag models according to Source 8.

(k) For the compilation in `Eclipse` create new make targets for the phase and interfacial force models, respectively.

- Click onto the `interfacialModels` in the *make targets* tab. In order to compile them as libraries – in `OpenFOAM` this would be done by typing `wmake libso` – name your target `libso` and don't change the default builder settings `wmake`.
- In the same way create `wclean` targets for your classes. Finally your make targets should look as illustrated in Figure 8.
- First compile the libraries and then compile the complete solver again. Do not forget to save all the files you have changed before.

At this point you may take advantage of one of `Eclipse`' powerful features - the fully linked error and warning marks. Selecting the *Problems* tab on the bottom lists all errors and warnings. Clicking on them opens the appropriate files where the error or warning is marked on the left margin.

```
Info<< "Reading transportProperties\n" << endl;

IOdictionary transportProperties
(
    IOobject
    (
        "transportProperties",
        runTime.constant(),
        mesh,
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    )
);

autoPtr<phaseModel> phasea = phaseModel::New
(
    mesh,
    transportProperties,
    "a"
);

autoPtr<phaseModel> phaseb = phaseModel::New
(
    mesh,
    transportProperties,
    "b"
);


const dimensionedScalar& rhoa = phasea->rho();
const dimensionedScalar& nua = phasea->nu();
const dimensionedScalar& da = phasea->d();

const dimensionedScalar& rhob = phaseb->rho();
const dimensionedScalar& nub = phaseb->nu();
const dimensionedScalar& db = phaseb->d();

dimensionedScalar Cvm
(
    transportProperties.lookup("Cvm")
);

dimensionedScalar Cl
(
    transportProperties.lookup("Cl")
);

dimensionedScalar Ct
(
    transportProperties.lookup("Ct")
);
```

Source Code 5: createFields.H

```
    IOdictionary interfacialProperties
    (
        IOobject
        (
            "interfacialProperties",
            runTime.constant(),
            mesh,
            IOobject::MUST_READ,
            IOobject::NO_WRITE
        )
    );

    autoPtr<dragModel> draga = dragModel::New
    (
        interfacialProperties,
        alpha,
        phasea,
        phaseb
    );

    autoPtr<dragModel> dragb = dragModel::New
    (
        interfacialProperties,
        beta,
        phaseb,
        phasea
    );
```

Source Code 6: createFields.H

```
/*-------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 1991-2008 OpenCFD Ltd.
     \\/     M anipulation  |
-------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software; you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by the
    Free Software Foundation; either version 2 of the License, or (at your
    option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
```

```
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM; if not, write to the Free Software Foundation,
    Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Application
    bubbleFoam

Description
    Solver for a system of 2 incompressible fluid phases with one phase
    dispersed, e.g. gas bubbles in a liquid.

\*---------------------------------------------------------------------------*/

#include "fvCFD.H"
#include "nearWallDist.H"
#include "wallFvPatch.H"
#include "Switch.H"
#include "dragModel.H"

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

Source Code 7: bubbleFoamMod.C

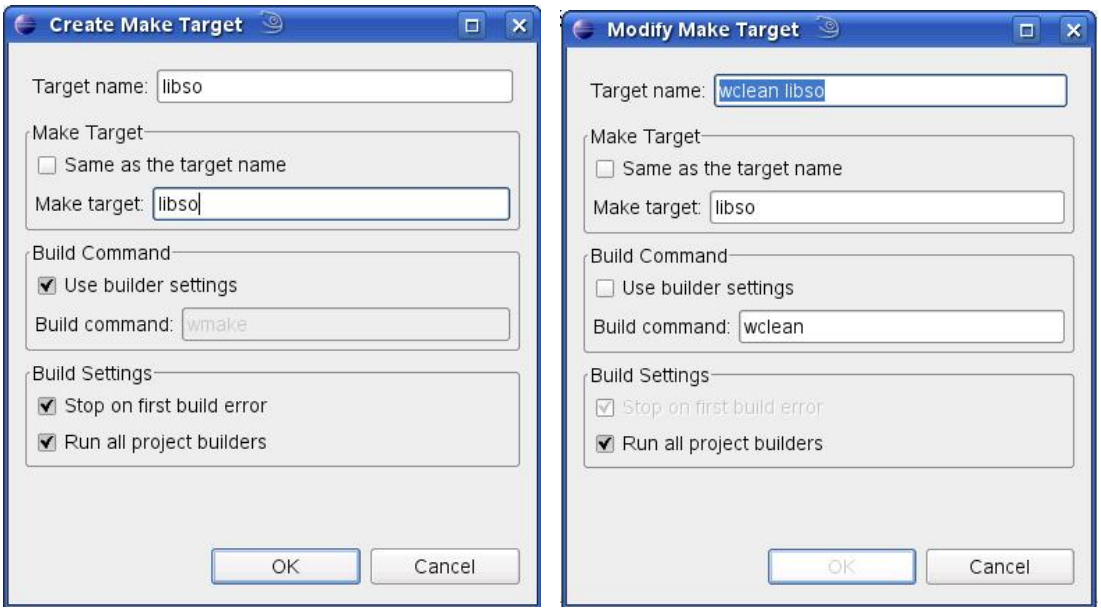
```
    volVectorField Ur = Ua - Ub;
    volScalarField magUr = mag(Ur);

    volScalarField Cda = draga->K(magUr);
    volScalarField Cdb = dragb->K(magUr);

    // corresponds to dragPhase == "blended" in twoPhaseEulerFoam
    volScalarField dragCoef =
    (
        "Cd",
        beta*Cda + alpha*Cdb
    );

    volVectorField liftCoeff = alpha*rhob*Cl*(Ur ^ fvc::curl(Ub));
```

Source Code 8: liftDragCoeffs.H



(a) C++ library make target **(b)** C++ library clean target

Figure 7: Eclipse make target for C++ libraries

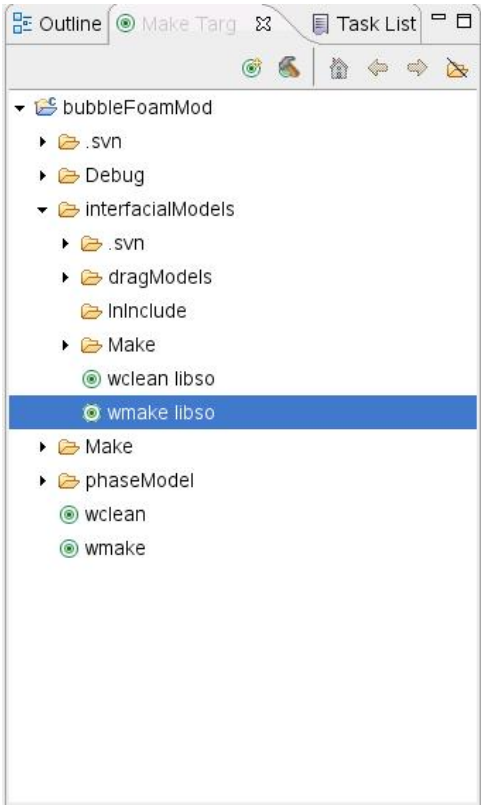


Figure 8: Overview: make targets

2.3 Implementing the models

After giving the solver the adequate C++ class base structure, new interfacial force models can be implemented easily.

1. To start with the implementation of the Tomiyama drag model create the new directory `Tomiyama98` in `bubbleFoamMod/interfacialModels/dragModels` and copy all files from `.../dragModels/SchillerNaumann` in the new subdirectory.
2. Rename (F2) to `Tomiyama98.C` and `Tomiyama98.H`, respectively.
3. Add the new model to the existing solver in the `bubbleFoamMod/interfacialModels/Make/files` file by adding the following line: `dragModels/Tomiyama98/Tomiyama98.C`.
4. For the implementation of a *new* drag force model, change the equations accordingly to the Tomiyama drag force model as given in Source 9 and 10, respectively. For the change of name you can use *Edit* → *Find/Replace*.
5. For the calculation of the Eötvös number the surface tension and gravity (as `dimensionedScalars`) are required. In order to make this properties available, some modifications to the phase model have to be made. Therefore read in the surface tension `sigma` and gravity `g` from a dictionary entry (see `phaseModel.C` – Source 11, lines 57 to 64). Add the lines 68 to 72 and lines 126 to 134 (access functions of `sigma` and `g`) in `phaseModel.H` as shown in Source 12.
6. Create make targets for the `phaseModel` library as done for the `interfacialModels` and compile it. Then recompile the `interfacialModels` library and at last the `bubbleFoamMod` solver.

```
/*------------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 1991-2007 OpenCFD Ltd.
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software; you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by the
    Free Software Foundation; either version 2 of the License, or (at your
    option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM; if not, write to the Free Software Foundation,
    Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

\*------------------------------------------------------------------------------*/

#include "Tomiyama98.H"
```

```
#include "addToRunTimeSelectionTable.H"

// * * * * * * * * * * * * * * Static Data Members * * * * * * * * * * * * * //
namespace Foam
{
    defineTypeNameAndDebug(Tomiyama98, 0);

    addToRunTimeSelectionTable
    (
        dragModel,
        Tomiyama98,
        dictionary
    );
}

// * * * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * //
Foam::Tomiyama98::Tomiyama98
(
    const dictionary& interfaceDict,
    const volScalarField& alpha,
    const phaseModel& phasea,
    const phaseModel& phaseb
)
:
    dragModel(interfaceDict, alpha, phasea, phaseb)
{}

// * * * * * * * * * * * * * * * * * Destructor  * * * * * * * * * * * * * * //
Foam::Tomiyama98::~Tomiyama98()
{}

// * * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * * * //
Foam::tmp<Foam::volScalarField> Foam::Tomiyama98::K
(
    const volScalarField& Ur
) const
{
    volScalarField Re = max(Ur*phasea_.d()/phaseb_.nu(), scalar(1.0e-3));
    volScalarField Eo = (phaseb_.rho()-phasea_.rho())*phasea_.g()*
                        pow(phasea_.d(),2)/phaseb_.sigma()*Re/Re;

    // Tomiyama Correlation for contamined systems
    volScalarField Cds = 24.*(scalar(1) + 0.15*pow(Re, 0.687))/Re;
    volScalarField Cdj = 8./3.*(Eo/(Eo+4.));

    forAll(Cds, celli)
    {
        if(Cds[celli] < Cdj[celli])
        {
            Cds[celli] = Cdj[celli];
        }
    }

    return 0.75*Cds*phaseb_.rho()*Ur/phasea_.d();
}
```

Source Code 9: Tomiyama98.C

```
/*———————————————————————————————————————————————————————————————————*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 1991–2007 OpenCFD Ltd.
     \\/     M anipulation  |
———————————————————————————————————————————————————————————————————

License
    This file is part of OpenFOAM.

    OpenFOAM is free software; you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by the
    Free Software Foundation; either version 2 of the License, or (at your
    option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM; if not, write to the Free Software Foundation,
    Inc., 59 Temple Place, Suite 330, Boston, MA 02111–1307 USA

Class
    Tomiyama98

Description

SourceFiles
    Tomiyama98.C

\*———————————————————————————————————————————————————————————————————*/

#ifndef Tomiyama98_H
#define Tomiyama98_H

#include "dragModel.H"
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
namespace Foam
{

/*———————————————————————————————————————————————————————————————————*\
                           Class Tomiyama98 Declaration
\*———————————————————————————————————————————————————————————————————*/

class Tomiyama98
:
    public dragModel
{

public:

    //– Runtime type information
    TypeName("Tomiyama98");


    // Constructors
```

```
        //- Construct from components
        Tomiyama98
        (
            const dictionary& interfaceDict,
            const volScalarField& alpha,
            const phaseModel& phasea,
            const phaseModel& phaseb
        );

    // Destructor
        ~Tomiyama98();


    // Member Functions
        tmp<volScalarField> K(const volScalarField& Ur) const;
};

} // End namespace Foam

#endif
```

Source Code 10: Tomiyama98.H

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright held by original author
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software; you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by the
    Free Software Foundation; either version 2 of the License, or (at your
    option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM; if not, write to the Free Software Foundation,
    Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

\*---------------------------------------------------------------------------*/
// * * * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * * * * * * //
    g_
    (
        dict_.lookup("g")
    ),
    sigma_
    (
```

```
        dict_.lookup("sigma")
    ),
```

Source Code 11: phaseModel.C

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright held by original author
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software; you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by the
    Free Software Foundation; either version 2 of the License, or (at your
    option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM; if not, write to the Free Software Foundation,
    Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Class
    Foam::phaseModel

SourceFiles
    phaseModel.C

\*---------------------------------------------------------------------------*/
/*---------------------------------------------------------------------------*\
                           Class phaseModel Declaration
\*---------------------------------------------------------------------------*/
        //- gravity
        dimensionedScalar g_;

        //- surface tension
        dimensionedScalar sigma_;
    // Member Functions
        const dimensionedScalar& g() const
        {
            return g_;
        }

        const dimensionedScalar& sigma() const
        {
            return sigma_;
        }
```

Source Code 12: phaseModel.H

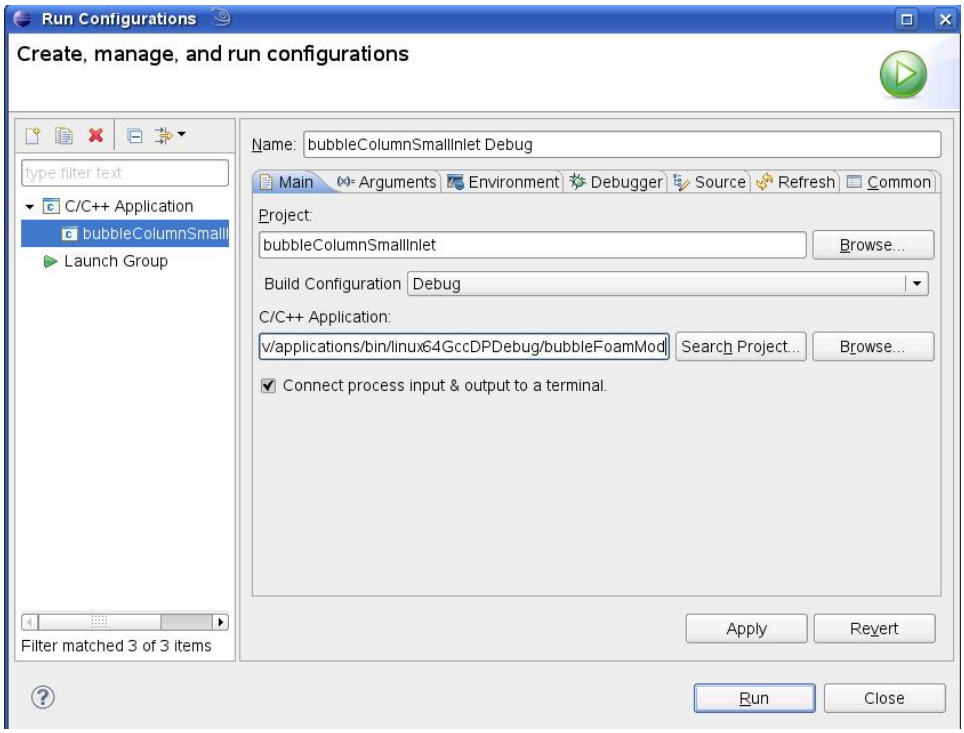


Figure 9: Run Configurations

3 Adjusting the case files

A simulation run can be accomplished within `Eclipse`, too.

1. Import the provided test case as a `C++` project and adapt the project properties for OpenFOAM as mentioned in chapter 2.
2. The new drag force model can be selected via keyword entry in the dictionary `interfacialProperties`, while the new transport properties have to be added in the `transportProperties` dictionary.
3. Set up the run configurations in the menu bar under *Run* → *Run Configurations*. Double-click on *C/C++ Application*, choose your test case as project and your solver as C/C++ application. Here, the test case `cylindricBubbleColumn` is chosen as project and the solver `bubbleFoamMod` is chosen as application – be aware to select the correct binary file (see Figure 9).
4. Start your simulation by clicking on the `Run` button and have a look at the console output in `Eclipse`.

4 Debugging with Eclipse

One of `Eclipse` capabilities is efficient debugging. So, in this chapter we would like to check if the new model is implemented correctly and how the runTime-selective access to the models is carried out.

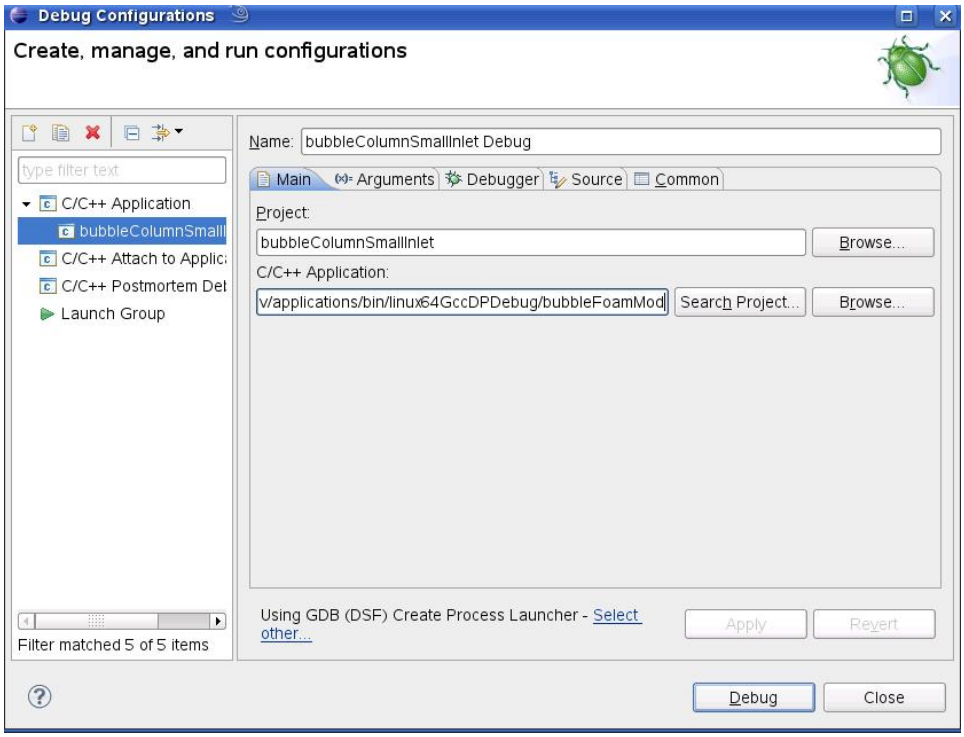


Figure 10: Debug Configurations

1. Set up the debugging configurations in *Run* → *Debug Configurations*. Choose the same settings for project and application as for the run configurations.
2. Make sure, that you are using the GDB (DSF) Create Process Launcher as debugger. If necessary, change it by clicking on *Select other...* (see Figure 10).
3. Start clicking on *Debug*. Now, the debug perspective should open - if not, activate it in the menu bar under *Window* → *Open Perspective* → *Debug*. The program should stop at the first breakpoint that is default set to entering `main{}`.

In Debug mode `Eclipse` allows you to set breakpoints by double-clicking on the bar to the left of the line numbers. You can resume debugging by clicking on the green play button (short cut *F8*). During debugging the console output on the bottom as well as the variable values and breakpoints on the right hand side are available (see Figure 11). You can walk through the code line-by-line while `Eclipse` highlights the line of the file you are currently computing. For stepping into function use the *step-into*-button (short cut *F5*) and for stepping over functions use the corresponding *step-over*-button (short cut *F6*).

4. Set the first breakpoint in `createFields.H` where you implemented the selection of the drag model (`autoPtr<dragModel> draga = dragModel::New`). Set the second breakpoint in `liftDragCoeffs.H` where `Cds` is evaluated using the new drag model (`volScalarField Cda = draga->K(magUr)`).
5. Now resume to the first breakpoint (*F8*) and step into the drag model selection process (*F5*). The debugger now points to the constructor of the new drag model where the drag model to be chosen is read in from the dictionary entry in `interfacialProperties`.
6. Resume to the next breakpoint (*F8*) and step into (*F5*) the evaluation of the drag coefficient. Depending on the model chosen in the `interfacialProperties` dictionary

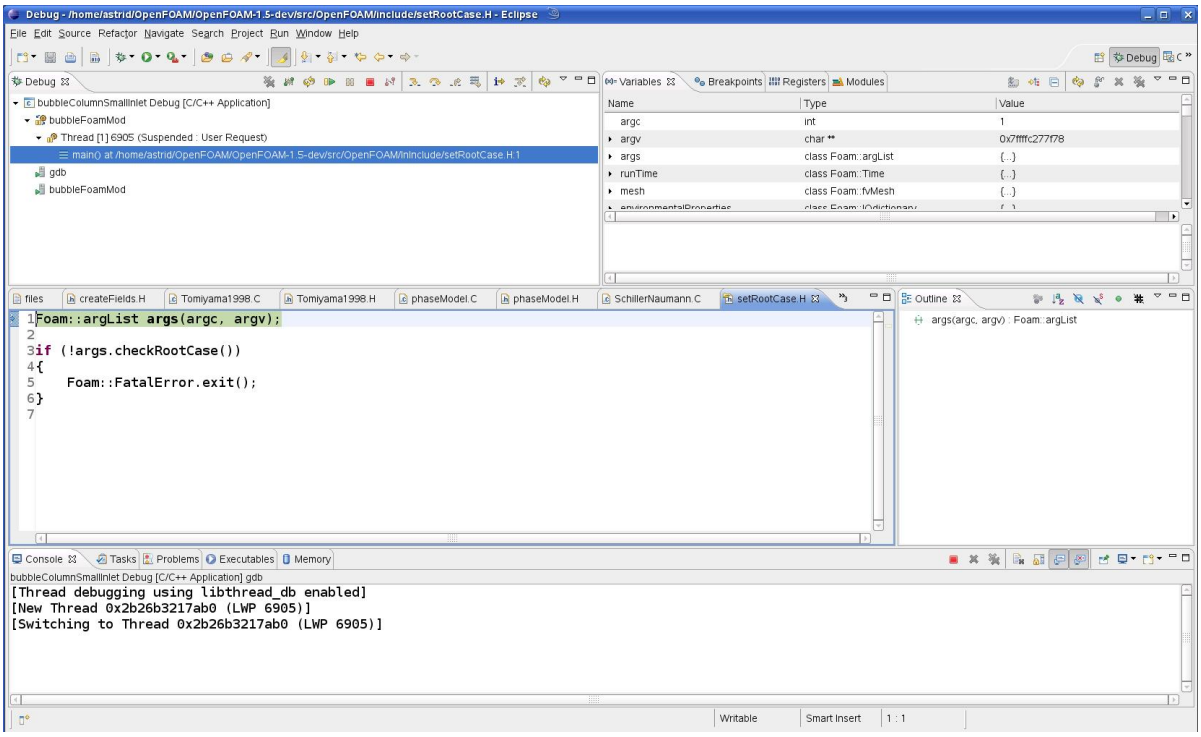


Figure 11: Debug Perspective

the debugger now points to either SchillerNaumann.C or Tomiyama98.C - for both cases calling the function `K(magUr)`. Step into (*F5*) the evaluation of the Reynolds number

```
volScalarField Re = max(Ur*phasea_.d()/phaseb_.nu(), scalar(1.0e-03));
```

and make sure that the bubble diameter of phase a (`phasea_.d()`) is accessed via access function of `phaseModel`. You can even check the value of the diameter variable in the *Variables* window on the right side.

A References

References

[1] Cdt wiki. available online at `http://wiki.eclipse.org/index.php/CDT`, March 2010.

[2] Eclipse homepage. available online at `http://www.eclipse.org`, March 2010.

[3] Eclipse tutorial. available online at `http://www.cs.umanitoba.ca/~eclipse/7-EclipseCDT.pdf`, March 2010.

[4] C. Mendez, N. Nigro, and A. Cardona. Drag and non-drag force influences in numerical simulations of metallurgical ladles. *Journal of Materials Processing Technology*, 160:296–305, 2005.

[5] H. Rusche. *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. PhD thesis, Imperial College of Science, Technology and Medicine, London, 2002.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "**publisher**" means any person or entity that distributes copies of the Document to the public.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.