Special Interest Group (SIG)
on Multiphase Flows in OpenFOAM®

Tutorial

Particles with pyFoam – The air classification test case

Astrid Mahrla* and Bernhard Gschaider

***Technische Universität München, Email: mail@amahrla.de**

Report version 1.0

July 2011

Contents

Preamble

This document was typeset with LATEX 2ε. Mathematical symbols were realised using the AMS® packages for LATEX provided by the American Mathematical Society.

GNU Free Documentation License

Version 1.3, 3 November 2008

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.

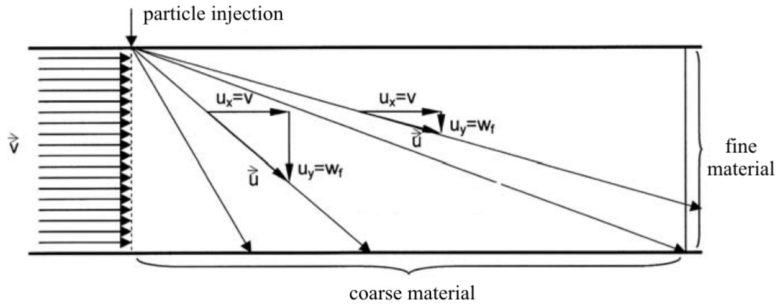


Figure 1: Stationary particle movement in horizontal fluid flow - principle of a cross-flow air classifier[1]

1 Introduction

1.1 Air Classification

Air Classifiers are commonly employed in technical industry where partly large product streams need to be seperated into two or more particle sizes. The separation criterion, the sinking velocity, results from surface and body forces affecting a particle in a fluid. Forces affecting the surface of a particle as pressure and drag forces compete with body forces affecting the particle's mass as gravity, centrifugal or inertia forces.

In this case a cross-flow air classifier is simulated, where the resulting velocity of the particle $\mathbf{u}$ is obtained by vector addition of the flow velocity of the fluid $\mathbf{v}$ and the relative velocity between particle and fluid $\mathbf{w}$:

$$\mathbf{u} = \mathbf{v} + \mathbf{w} \tag{1}$$

The affecting force in horizontal direction is drag force, while the affecting force in vertical direction includes gravity, buoyancy and drag. In the special case of horizontal flow velocity, particles with lower sinking velocity due to the size of the particles or a lower difference in density between particle and fluid are taken further in fluid flow direction than particles with higher sinking velocity (see Figure 1).

From this, one obtains the particle distribution of a particle collective[2]. In general, a particle class is described by a range of a particle characteristic as, for instance, the particle diameter x. Using a histogram for the size distribution the particle distribution density $q_r(x)$ is shown on the ordinate and plotted versus particle size on the abscissa. Then area of a column $q_{r,i} \cdot \Delta x_i$ equates to the particle amount $\frac{\Delta m_i}{m_{tot}}$ (see Figure 2).

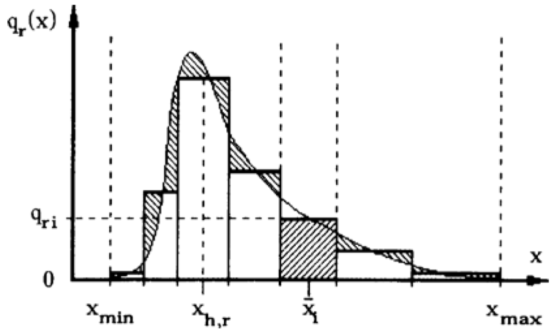


Figure 2: Particle distribution density[1]

1.2 Simulation of gas-solid particle flows using Euler-Lagrange approach

For the simulation of gas-solid particle flows the Euler-Lagrange approach is used here. The gas phase is treated with Eulerian method, so the incompressible Navier-Stokes equations are applied for the continuous phase. The momentum source term S_p on the right hand side of equation 2 includes the effect of the particles on the fluid phase.

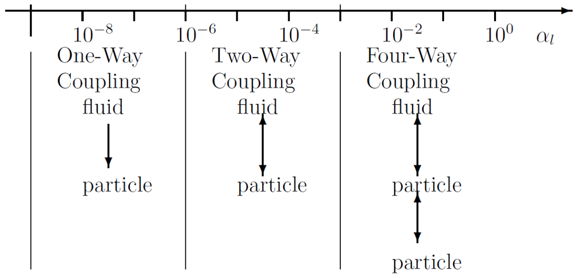
$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \bullet \rho \mathbf{U}\mathbf{U} = -\nabla p + \nabla \bullet (\mu(\nabla \mathbf{U} + \nabla \mathbf{U}^{\mathrm{T}})) + \rho g + S_p \tag{2}$$

In contrast, the particles are simulated in a Lagrangian way. Therefore, they are tracked on a grid using the differential equations for particle motion that are based on Newton's 2nd law. The net force acting on each individual particle is calculated considering all the relevant forces. Forces acting on each particle as gravity or drag force are summed up as $\mathbf{F}_p$, forces due to particle-particle interaction are summarized as $\mathbf{F}_c$[2]. (Eq. 3 and 4)

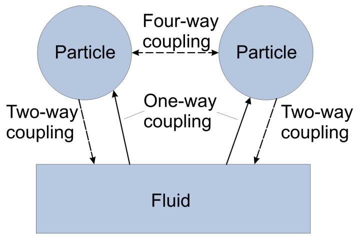
$$\frac{\partial \mathbf{X}_p}{\partial t} = \mathbf{U}_p \tag{3}$$

$$m_p \frac{\partial \mathbf{U}_p}{\partial t} = \sum \mathbf{F}_p + \sum \mathbf{F}_c \tag{4}$$

In general, if more than one phase is simulated, the interaction of the phases is considered by coupling. These effects strongly depend on the particle phase fraction. One-way coupling is given when the fluid affects the discrete phase through forces as drag force (Eq. 4). If there is an additional reverse effect, the flow is two-way coupled. This is gained by the implementation of a momentum source term in the continuous phase equation (Eq. 2).



(a) Dependency of coupling on particle phase fraction[3]



(b) Coupling between gas and solid phase

Figure 3: Phase coupling of gas-solid flows

Three- and four-way coupling respectively is included if particle-particle interaction are taken into account. One way of modelling inter-particle forces are collision models, where an impact is described as a combination of mechanical forces acting on the colliding particles (Eq. 4). (see Fig. 3)

The explicit coupled solution algorithm for both phases in the Eulerian-Lagrangian framework is shown in fig. 4. At the beginning of the simulation data for the grid as well as boundary and initial conditions for the gas phase are imported and the particles are initialized. At the beginning of the time step Δt the continuous flow properties are calculated due to Finite Volume Method (FVM). Secondly, the individual motion calculation for the particles is executed in regard to the previously determined conditions of the continuous flow calculation. For this step sub-time stepping is available and mostly necessary if collision calculation is performed. After the particle motion the momentum source term for the gas phase is calculated and updated for the continuous phase calculation in the next time step - as long as the end time of the simulation is not reached.

1.3 Implementation of the particle class in rhoPisoAbscheidParcelFoam15

The `rhoPisoAbscheidParcelFoam15` solver is a two-way coupled gas-solid flow solver. The particle class in OpenFOAM is implemented as an particle-cloud system, where the particle properties as position, diameter, mass or velocity are stored as data elements of a

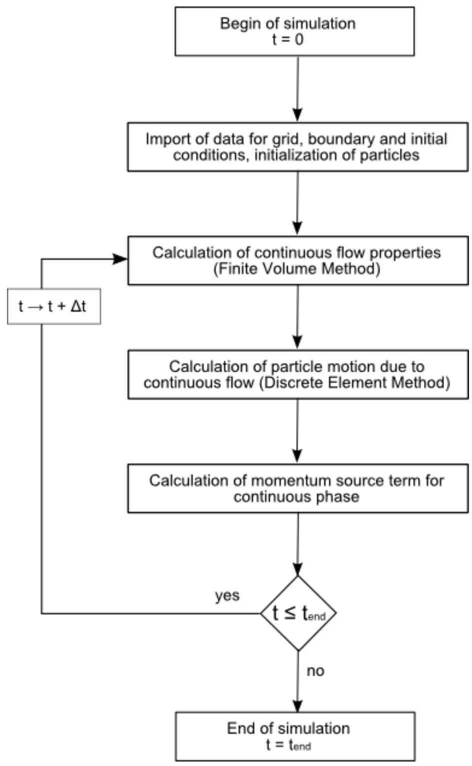


Figure 4: Explicit coupled solution algorithm

particle. The appropriate *cloud* integrates the attributes of the particles in a computational cell, e. g. the momentum source term introduced from all particles in a cell or the fluid flow properties of the gas phase in a certain cell.

Herein, we use `basicAbscheidCloud` as *cloud* and `basicAbscheidParcel` as *particle* class. In order to use them in top-level coding, we include the `basicAbscheideCloud.H` file in the header of our top-level code file `rhoPisoAbscheidParcelFoam.C` in line 37. At the beginning of the `main` part the initialization of the cloud is called using `#include "createClouds.H"` in line 49. During the initialization in the `createClouds.H` file, the object `kinematicCloud1` of the `basicAbscheideCloud` class is created. During the time loop the particles are moved calling `kinematicCloud1.evolve()` in line 71 and some information are passed back using `kinematicCloud1.info()` in line 72. Afterwards the continuous flow field is calculated solving the momentum equation, which is defined in `UEqn.H`, in line 80.

```
pointMesh pMesh(mesh);
volPointInterpolation vpi(mesh, pMesh);

Info<< "Constructing abscheideCloud1" << endl;
basicAbscheideCloud kinematicCloud1
(
    "kinematicCloud1",
```

```
        vpi,
        rho,
        U,
        thermo->mu(),
        g
    );
```

Source Code 1: createClouds.H

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 1991-2009 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
    rhoPisoTwinParcelFoam

Description
    Transient solver for compressible, turbulent flow with two thermo-clouds.

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

#include "fvCFD.H"
#include "basicThermo.H"
#include "compressible/RASModel/RASModel.H"

#include "basicAbscheideCloud.H"

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

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

```

```
#include "createTime.H"
#include "createMesh.H"
#include "readEnvironmentalProperties.H"
#include "createFields.H"
#include "createClouds.H"
#include "readPISOControls.H"
#include "initContinuityErrs.H"
#include "readTimeControls.H"
#include "compressibleCourantNo.H"
#include "setInitialDeltaT.H"

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

Info<< "\nStarting time loop\n" << endl;

while (runTime.run())
{
    #include "readTimeControls.H"
    #include "readPISOControls.H"
    #include "compressibleCourantNo.H"
    #include "setDeltaT.H"

    runTime++;

    Info<< "Time = " << runTime.timeName() << nl << endl;

    kinematicCloud1.evolve();
    kinematicCloud1.info();


    #include "rhoEqn.H"

    // --- PIMPLE loop
    for (int ocorr=1; ocorr<=nOuterCorr; ocorr++)
    {
        #include "UEqn.H"

        // --- PISO loop
        for (int corr=1; corr<=nCorr; corr++)
        {
            #include "hEqn.H"
            #include "pEqn.H"
        }
    }

    turbulence->correct();

    rho = thermo->rho();

    runTime.write();

    Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
        << "  ClockTime = " << runTime.elapsedClockTime() << " s"
        << nl << endl;
}
```

```
    Info<< "End\n" << endl;

    return 0;
}


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

Source Code 2: rhoPisoAbscheidParcelFoam.C

Having a closer look at the momentum equation in `UEqn.H`, one will find the momentum source term `kinematicCloud1.SU1()` due to particle-fluid interaction in line 7.

```
1     fvVectorMatrix UEqn
2     (
3         fvm::ddt(rho, U)
4       + fvm::div(phi, U)
5       + turbulence->divDevRhoReff(U)
6      ==
7         kinematicCloud1.SU1()
8       + rho.dimensionedInternalField()*g
9     );
10
11    UEqn.relax();
12
13    if (momentumPredictor)
14    {
15        solve(UEqn == -fvc::grad(p));
16    }
```

Source Code 3: UEqn.H

As the `basicAbscheideCloud` and the `basicAbscheideParcel` are inherited from the `KinematicCloud` and `KinematicParcel` classes, they possess the same properties as the parent classes. They use all `subModels` as dispersion, drag or injection models of the `Kinematic` class. Each submodels is included into the derived class as can be seen in `makeBasicAbscheideParcelDispersionModels.C` for the dispersion models.

For the air classifier test case we should detect where the particles are deposited, e.g. which patch a particle touches. Therefore in the `basicAbscheideParcel.C` file an info output is produced, when accessing the `hitPatch` function in line 26.

```
1  /*---------------------------------------------------------------------------*\
2    =========                 |
3    \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
4     \\    /   O peration     |
```

```
    \\  /    A nd           | Copyright (C) 1991-2009 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

\*---------------------------------------------------------------------------*/
void Foam::basicAbscheideParcel::hitPatch
(
    const polyPatch& p,
    trackData& td
)
{
    Pout << "Abgeschieden at " << p.name() << " with m: " << mass() <<
                " N: " << nParticle() << " = total: " << mass()*nParticle() <<
endl;

    KinematicParcel<basicAbscheideParcel>::hitPatch(p,td);
```

Source Code 4: basicAbscheideParcel.C

For further information on the underlying particle class, you will find the source code in the OpenFoam folder `src/lagrangian/intermediate`. Preferibly analyze the source code in Eclipse or an alternative Integrated Development Environment (IDE), use the native `rhoPisoTwinParcelFoam` solver and `simplifiedSiwek` case that can be found in `tutorials/lagrangian/rhoPisoTwinParcelFoam`.

1.4 Test case

The air classifier test case consists of a horizontal channel with two cavities for the classification of the particles. The particles are injected conoidally on the left side of the geometry.

In the `kinematicCloud1Properties` dictionary all properties concerning the particles are declared. For instance the particle models as the injection model, drag model or dispersion model are chosen. The `coupled` switch is set to `true`, so the momentum source term for

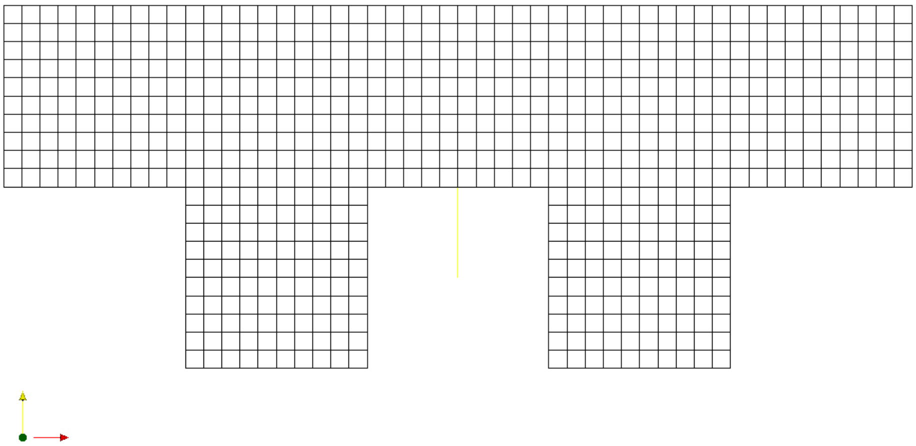


Figure 5: Mesh: air classifier

the particles will be evaluated. Additionally mass, density and particle size distribution of the particles are defined. The interpolation schemes for the continuous fluid phase for the calculation of particle motion are set.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      kinematicCloud1Properties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

InjectionModel  ConeInjection;

DragModel       SphereDrag;

DispersionModel StochasticDispersionRAS;

WallInteractionModel StandardWallInteraction;

PostProcessingModel none;

coupled         true;

cellValueSourceCorrection on;

```

```
parcelTypeId    2;

rhoMin          rhoMin [ 1 -3 0 0 0 ] 1e-15;
minParticleMass minParticleMass [ 1 0 0 0 0 ] 1e-15;
rho0            rho0 [ 1 -3 0 0 0 ] 5000;

interpolationSchemes
{
    rho             cell;
    U               cellPoint;
    mu              cell;
}

integrationSchemes
{
    U               Euler;
}

particleForces
{
    gravity         on;
    virtualMass     off;
    pressureGradient off;
}

NoInjectionCoeffs
{
    SOI             0.001;
}

ConeInjectionCoeffs
{
    SOI 1;
    duration 0.5;
    parcelsPerSecond 1000;
    position        ( 0.01 0.05 0 );
    direction       ( 1 1 0 );
    volumeFlowRate  Constant;
    volumeFlowRateCoeffs
    {
        value                           1;
    }
    Umag            Constant;
    UmagCoeffs
    {
        value                           1;
    }
    thetaInner      Constant;
    thetaInnerCoeffs
    {
        value                           0.0;
    }
    thetaOuter      Constant;
    thetaOuterCoeffs
    {
        value                           30.0;
```

```
    }

    parcelPDF
    {
        pdfType         uniform;
        RosinRammlerPDF
        {
            minValue        5e-05;
            maxValue        0.0001;
            d               ( 7.5e-05 );
            n               ( 0.5 );
        }
        uniformPDF
        {
            minValue        5e-05;
            maxValue        0.0001;
        }
    }
}

StandardWallInteractionCoeffs
{
    e               e [ 0 0 0 0 0 ] 1;
    mu              mu [ 0 0 0 0 0 ] 0;
}

parcelBasisType mass;
massTotal m [1 0 0 0 0 0 0] 0.01;

PatchInjectionCoeffs
{
    SOI 1;
    duration 0.5;
    parcelsPerSecond 1000;
    U0 (1 0 0);
    volumeFlowRate constant 1;
    parcelPDF
    {
        pdfType         uniform;
        RosinRammlerPDF
        {
            minValue        5e-05;
            maxValue        0.0001;
            d               ( 7.5e-05 );
            n               ( 0.5 );
        }
        uniformPDF
        {
            minValue        5e-05;
            maxValue        0.0001;
        }
    }
    patchName inlet;
}

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

Source Code 5: kinematicCloud1Properties

2 Postprocessing and parameter variation for apparatus optimization with PyFoam

2.1 Scripting with Python

In this chapter we would like to present the opportunities in using the Python binding for scripting in OpenFOAM: PyFoam[4]. This Python[5] library can be used to

- analyze the log files produced by OpenFOAM
- execute the parameter files and the initial conditions of a simulation
- plot residuals of OpenFOAM solvers

For further information on PyFoam, have a look at the presentation "Happy Foaming with Python" from the 4th OpenFOAM workshop [6] or the presentation from the 5th OpenFOAM workshop on "Automatization with pyFoam"[7].

2.2 PyFoam Installation

The installation will follow the OpenFOAM wiki [4]. Download the latest version of PyFoam - we use PyFoam-0.5.3 and follow the installation instruction for the installation as root as described in chapter 3 of the wiki entry. Copy the .tar-file to `user-1.5/applications/utilities/` and extract the file using `tar -xzf PyFoam-0.5.3.tar.gz`. Go into the PyFoam folder and login as root typing `su`. Install PyFoam: `python setup.py install`. Don't forget to logout as root. Test the installation as described on the OpenFOAM wiki.

2.3 Running the test case with PyFoam

Compile the solver `rhoPisoAbscheidParcelFoam15` and create the mesh of the `dreiAbscheidCavities15` case using `blockMesh`. Then go into the `Scripts` folder and execute `runAndAnalyzeSolver.py` using `python runAndAnalyzeSolver.py`

`../dreiAbscheidCavities15`. This Python scripts runs your test case automatically and summarizes your particle data.

```
#! /usr/bin/python

import re,sys

from PyFoam.LogAnalysis.LogLineAnalyzer import LogLineAnalyzer
from PyFoam.LogAnalysis.BoundingLogAnalyzer import BoundingLogAnalyzer
from PyFoam.Execution.AnalyzedRunner import AnalyzedRunner

class AbscheideAnalyzer(LogLineAnalyzer):
    def __init__(self):
        LogLineAnalyzer.__init__(self)

        self.massLeft=0
        self.removed={}

        self.removedExpr=re.compile("Abgeschieden at (.+) with m: (.+) N: (.+)
                                    = total: (.+)")
        self.massExpr=re.compile("Current mass in system = (.+)")

    def analyzeLine(self,line):
        m=self.removedExpr.match(line)
        if m!=None:
            name=m.groups()[0]
            mass=float(m.groups()[1])
            nr=float(m.groups()[2])
            total=float(m.groups()[3])

            try:
                self.removed[name]+=total
            except KeyError:
                self.removed[name] =total

        m=self.massExpr.match(line)
        if m!=None:
            self.massLeft=float(m.groups()[0])

    def doReport(self):
        summe=self.massLeft
        print "Mass left in system:",self.massLeft
        for k,v in self.removed.iteritems():
            print "Removed by",k,":",v
            summe+=v
        print "Total mass accounted for:",summe

    def addTimeListener(self,other):
        pass

case=sys.argv[1]

abscheid=AbscheideAnalyzer()

run=AnalyzedRunner(abscheid,
```

```
53                     argv=["rhoPisoAbscheidParcelFoam","-case",case],
54                     silent=True)
55 run.start()
56
57 abscheid.doReport()
```

Source Code 6: runSolverAndAnalyze.py

In the first lines of `runSolverAndAnalyze.py` the system libraries `re` and `sys` as well as the `PyFoam` libraries for the analyzation of the generated log file are imported. In ll. 9 a new class `AbscheideAnalyzer` is defined by inheritation of the `LogLineAnalyzer` class. This class observes the output produced from the `hitPatch` function in `basicAbscheideParcel` class (see basicAbscheideParcel.C, ll. 26) and registers every time a particle leaves through a patch and performs mass evaluation. The `analyzeLine` method (ll. 20) checks whether one of the expressions generated from `hitPatch` was matched and updates data of left particles at the respective patch and mass in current system. In ll. 50 an `AbscheideAnalyzer` object is created and simulation run and report are accomplished. [7]

For a first impression of the post processing, have a look at your case in paraview. The OpenFOAM parser `paraFoam` cannot view Lagrangian particles, so the Lagrangian data needs to be converted to the `VTK` format using `foamToVTK`. Then start paraview using the `paraview` command. Now, in paraview open the `.vtk`-file of your test case, by clicking on `Open File`, open the VTK folder of your test case and the `dreiAbscheidCavities15_..vtk` file in the `VTK` folder. This reads in your Eulerian data. Click the apply button and reduce the opacity in the `display` tab to have a look at your mesh and Eulerian data. Now open the Lagrangian data opening the `kinematicCloud1_..vtk` file in `VTK/lagrangian/kinematicCloud1` and confirm with the apply button.

Create glyphs to visualize your particles. Set the glyph type to sphere and the scale mode to scalar and make sure, that the scale factor is set to something like 20. Your visualized data should look like Fig. 6.

One of PyFoam's strengths is running test cases automatically. For the air classifier test case the inlet velocity is varied in line 20 of the `inletVelocityVariation.py` script. Run the script using `python inletVelocityVariation.py ../dreiAbscheidCavities15 inletVariation`. Additionally, an `inletVariation.cvs` file holding the particle data will be created. Continue evaluating this `inletVariaton.cvs` file.

The `inletVariation.py` script is based on `runAndAnalyze.py` supplemented by the function `addCSVLine` (see ll. 15) that adds data to a CSVCollector (a convenience-class for generating a consistent `csv`-file) and a `for` loop the the inlet velocity variation (see ll. 20). In the `for` loop the inlet velocity is set to defined values by changing the respective OpenFOAM file, this is done using the pyFoam library `ParsedParameterFile` that is

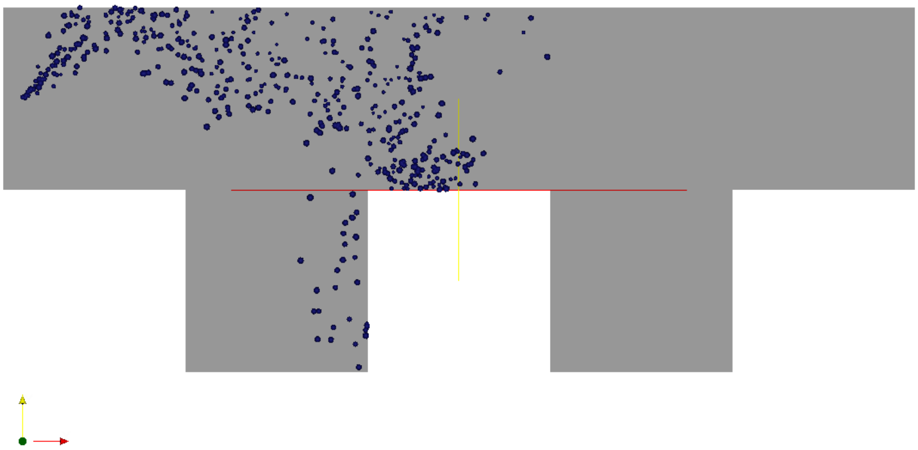


Figure 6: Visualization of particles in air classifier

capable editing entries for boundary and initial conditions. Therefore this library is included at the beginning of the script besides the `ClearCase` and `CSVCollection` libraries. [7]

```
#! /usr/bin/python

import re,sys
from os import path

from PyFoam.LogAnalysis.LogLineAnalyzer import LogLineAnalyzer
from PyFoam.LogAnalysis.BoundingLogAnalyzer import BoundingLogAnalyzer
from PyFoam.Execution.AnalyzedRunner import AnalyzedRunner
from PyFoam.Applications.ClearCase import ClearCase
from PyFoam.RunDictionary.ParsedParameterFile import ParsedParameterFile
from PyFoam.Basics.CSVCollection import CSVCollection

class AbscheideAnalyzer(LogLineAnalyzer):

    def addCSVLine(self,csv):
        csv["mass left in system"]=self.massLeft
        for k,v in self.removed.iteritems():
            csv[k]=v

for v in [0.1, 0.5, 1, 1.5,2,3]:
    print "Velocity set to",v
    ClearCase(args=["."])
    uInit=ParsedParameterFile(path.join(case,"0","U"))
    uInit["boundaryField"]["inlet"]["value"].setUniform([v,0,0])
    uInit.writeFile()
```

Source Code 7: inletVelocityVariation.py

3 Homework – Scripting with Python

Optimization of the inlet velocity

Based on `inletVelocityVariation.py` write your own Python script that automatically finds the optimal inlet velocity, so that as much as possible particles hit the second cavity.

Variation of another variable

Write a new Python script that varies another variable, for instance the direction of injection.

References

[1] Matthias Stieß. *Mechanische Verfahrenstechnik - Partikeltechnologie 1*. Springer, Berlin Heidelberg, 3rd edition, 2009.

[2] Clayton T. Crowe, Martin Sommerfeld, and Yutaka Tsuji. *Multiphase flows with droplets and particles*. CRC Press, Boca Raton, Fla. [u.a.], 1998.

[3] André Kaufmann. *Towards Eulerian-Eulerian Large Eddy Simulation of reactive two-phase flows*. PhD thesis, Insitut National Polytechnique de Toulouse, 2004.

[4] Pyfoam contribution. online available at `http://openfoamwiki.net/index.php/Contrib_PyFoam`, April 2011.

[5] Python programming language – official website. online availabe at `http://www.python.org`, April 2011.

[6] B.F.W. Gschaider. pyfoam – happy foaming with python. online available at `http://www.openfoamworkshop.org/2009/4th_Workshop/0_Feature_Presentations/OFW4_2009_Gschaider_PyFoam.pdf`, April 2011.

[7] B.F.W. Gschaider. Automatization with pyfoam. available online at `http://web.student.chalmers.se/groups/ofw5/Advanced_Training/pyFoamAdvanced.pdf`, April 2011.