A workflow for the sustainable development of closure models for bubbly flows

Hänsch, S.; Evdokimov, I.; Schlegel, F.; Lucas, D.;

Originally published:

June 2021

Chemical Engineering Science 244(2021), 116807

DOI: https://doi.org/10.1016/j.ces.2021.116807

A workflow for the sustainable development of closure models for bubbly flows

S. Hänsch[a], I. Evdokimov, F. Schlegel, D. Lucas

Helmholtz-Zentrum Dresden-Rossendorf e.V., Institute of Fluid Dynamics, 01314 Dresden, Germany

[a] *Corresponding author: s.haensch@hzdr.de*

Abstract

Many years of research in developing closure models for polydisperse bubbly flows have produced a plethora of empirical and semi-empirical models. The continuous development and analysis of such models requires their constant validation with the steadily increasing number of validation cases in the literature.

In this paper we present a pipeline for the fully-automated analysis of OpenFOAM simulations using the Snakemake workflow management system. The pipeline is applied to an extensive collection of well-established validation cases for bubbly flows and allows the fast and efficient production of large amounts of results that are summarized in well-structured reports. An optional post-processing step introduces a fuzzy logic controller developed for the detailed analysis of these results by quantifying the agreement of the simulation with the available experimental data. It is demonstrated how such quantification enables the systematic evaluation of new closure models and contributes to a more sustainable model development.

Keywords:
Baseline, bubbly flow, OpenFOAM, workflow, artificial intelligence

1. Introduction

Understanding and predicting multiphase flows is crucial for the development of reliable, safe and efficient devices in the nuclear, chemical, energy and oil-and-gas industry. Computational fluid dynamics (CFD) has proven to be an indispensable tool that adds to a better understanding of multiphase flows on all relevant scales, from single bubbles to the flow in large industrial components. Much effort was focused on achieving predictive capabilities of CFD simulations during the last decades. Today, the Eulerian-Eulerian two-fluid methodology is widely used for analysing multiphase flows on the industrial component-scale. Low computational cost and great flexibility in terms of possible flow geometries have made it the state-of-the-art approach for multiphase flow analysis in the industry. The reliability of results depends on the availability of accurate closure models and their correct application. Although good predictions can often be obtained for single cases, there is an ongoing effort to reduce the empiricism of models in favour of more mechanistic formulations for a better accuracy and generality of these models.

Decades of developing new models for Eulerian-Eulerian CFD have produced a vast amount of literature on the subject, and a huge number of both models and validation cases [1, 2, 3]. For each required Euler-Euler closure a multitude of empirical or semi-empirical models exists in the literature from which the CFD engineer is free to choose. This accumulation of models makes it increasingly difficult to give best practice guidelines for the simulation of bubbly flows. Evaluating new models and comparing them against the plethora of available alternatives becomes an increasingly tedious endeavour.

However, the huge amount of data to be found in the literature offers the opportunity to apply workflow tools for a faster and more efficient processing of CFD data. In other fields of research, such as bio-informatics [4] and RNA sequencing in particular [5], the development of workflows has already proven to be fundamental for gaining a deeper understanding of large data sets. The enormous amount of CFD data produced from existing validation cases and the increasing number of closure models make a fast and efficient tool to process all this data a crucial part of the Euler-Euler model development in the future. Furthermore, such automation allows us to explore the possibility of including data science and artificial intelligence tools in the analysis and evaluation process of new Euler-Euler models.

In Section 2 of this paper we describe the current status of Eulerian-Eulerian CFD modeling of multiphase flows. We will review the conventional way of developing and testing new models and highlight the issues that arise from following such an approach. We then present the alternative ‘baseline‘ strategy [6] developed at Helmholtz-Zentrum Dresden-Rossendorf (HZDR), which aims at converging towards a fixed set of universal closure models, so-called baseline models. An overview of the current collection of validation cases for baseline model testing available at HZDR is given. In Section 3 we then present the Snakemake workflow, which was developed for the fully-automated pre-processing, running and post-processing of the extensive amount of OpenFOAM cases in our collection. Section 4 will apply fuzzy logic, a subset of artificial intelligence, for the evaluation of simulation results. We will describe the parts of a fuzzy logic controller, designed to infer from two error metrics an output metric that allows us to compare the performance of new Euler-Euler models with the baseline model set. Results of a demonstration case are then presented in Section 5 where we test a new lift force correlation against the current baseline model. We describe the performance of the workflow and present validation graphs of selected cases to demonstrate the fuzzy logic evaluation. An overview of fuzzy logic results for the entire case collection aims to demonstrate the potential of such artificial intelligence tools for a sustainable Euler-Euler model development. Conclusions and an outlook to where our future work is going is given in Section 6.

2. Background

2.1. Conventional Euler-Euler model development

The description of two-phase flows with the Eulerian-Eulerian methodology requires many closure models. Closure models need to describe all the detailed flow phenomena, which the usually coarse computational grid cannot resolve. For polydisperse bubbly flows a dozen such closures is needed: interfacial force closures, i.e. drag, shear lift, turbulent dispersion, virtual mass and wall lubrication forces [7], turbulence closures for the shear and bubble-induced turbulence of the liquid phase [8], and closures describing the polydispersity of bubbles, i.e. a population balance model, and models for both coalescence and breakup events [9]. A lot of different model options for each of these closures have emerged in the literature, each of which comes with a multitude of adjustable model parameters.

Table 1 lists a selection of such models suitable for the modeling of bubbly flows, part of which are provided by OpenFOAM [10] and by the *HZDR Multiphase Addon for OpenFOAM* [11]. Listed are five drag models, three lift models, three models for turbulent dispersion, two for vitual mass and four for wall lubrication, three different turbulence models for the liquid and three different models for bubble-induced turbulence, two different approaches for population balance modeling, five different coalescence and four different binary breakup models. These 34 options alone give a total of 129600 possible model combinations. This number becomes even more extensive considering all the models in the literature not yet included in the HZDR model library. Numerous model parameters inherent in every sub-model further add to the complexity. The description of polydisperse bubbly flows using the Euler-Euler approach is inarguably based on a highly complex set of intricately-linked models making the interpretation of CFD results extremely difficult.

Table 1: Closure models for polydisperse bubbly flow available with the *HZDR Multiphase Addon for OpenFOAM* [11]. The current baseline model set is indicated in bold text.

Required closures		Model options	Parameter
Interfacial forces	Drag	**Ishii and Zuber [12]**	-
		Schiller and Naumann [13]	-
		Tomiyama et al. [14]	-
		Tomiyama et al. [15]	-
		Tomiyama et al. [16]	-
	Shear lift	**Tomiyama et al. [17]**	-
		Legendre and Magnaudet [18]	-
		Moraga et al. [19]	-
	Turbulent dispersion	**Burns et al. [20]**	1
		Gosman et al. [21]	1
		Lopez de Bertodano [22]	1
	Virtual mass	**Crowe et al. [23]**	1
		Lamb [24]	-
	Wall lubrication	**Hosokawa et al. [25]**	-
		Antal et al. [26]	2
		Frank [27]	3
		Tomiyama [28]	1
Turbulence	Shear-induced	**k-omega-SST [29]**	12
		k-epsilon [30]	6
		k-omega [31]	5
	Bubble-induced	**Ma et al. [32]**	-
		Rzehak and Krepper [33]	-
		Sato and Sadatomi [34]	1
Poly-dispersity	Population balance	**Class method [35]**	-
		Ishii et al. [36]	-
	Coalescence	**Liao et al. [9]**	9
		Coulaloglou and Tavlarides [37]	2
		Lehr et al. [38]	2
		Luo [39]	2
		Prince and Blanch [40]	3
	Breakup	**Liao et al. [9]**	4
		Lehr et al. [38]	-
		Laakkonen et al. [41]	4
		Luo and Svendsen [42]	3

There is no consensus about the use of any of those closure models. In the literature usually a different set of models is chosen by each research group [43, 1, 7, 44]. Model constants tend to be tuned for each specific validation case. Despite a large number of publications on the model development for the last two decades little progress is seen towards Eulerian-Eulerian simulations with reliable predictive abilities [6].

2.2. The 'Baseline' Strategy

The aim of the baseline strategy at HZDR is to arrive at a single universal set of Eulerian-Eulerian two-fluid models that can predict bubbly flows in any flow configuration. Baseline models should offer a certain range of applicability regarding the local flow characteristics. The baseline model set is fixed, as are the model constants. For different flow situations the contribution of each of the phenomena listed in Table 1 will of course vary, but the baseline model set should predict such changes without any further tuning. As described by Lucas et al. [6] for the sake of generality the models chosen for each phenomena should preferably be mechanistic models that are based on local flow conditions rather than empirical models derived from observations made in a specific experiment. The baseline model set should produce convincing results for a large range of different validation cases in order to prove its predictive abilities for future cases.

Table 1 lists the current set of baseline models for polydisperse bubbly flow at HZDR in bold letters. Find the detailed equations for these models in Liao et al. [8]. Over the years the models listed here have proven to repeatedly produce good results for a variety of cases [9, 7, 8].

Figure 1 illustrates the general process of updating this fixed set of baseline models in case a new model has been developed or found in the literature. In a first step the new model needs to be analysed regarding the model requirements of the baseline model set. Various features need to be checked, such as a physical and mechanistic basis of the model, its generality or its derivation from advanced experiments. If the advantages of the new model become apparent, it will then be tested in a next step. This part of the process is essential as it aims to demonstrate an overall improvement of the CFD predictions for a collection of cases. This case collection contains an extensive selection of validation cases, which will be described in the next section. Generally, a new model will replace the current baseline model if an improvement for the majority of our validation cases can be demonstrated.

During the continuous model development we try to identify deficiencies of models, in which we should invest our efforts for further improvement.

As illustrated on the right hand side of the flow diagram in Figure 1 a model update can be considered even if overall case results do not improve. The advantages of a new model then need to outweigh the deterioration of results. The coupling of physical phenomena is complex making a straightforward model evaluation impossible at times. Worse results obtained with a better model can be caused by the interdependence of the current set of baseline models, where potential shortcomings of one model are possibly compensated by the tuning of another.

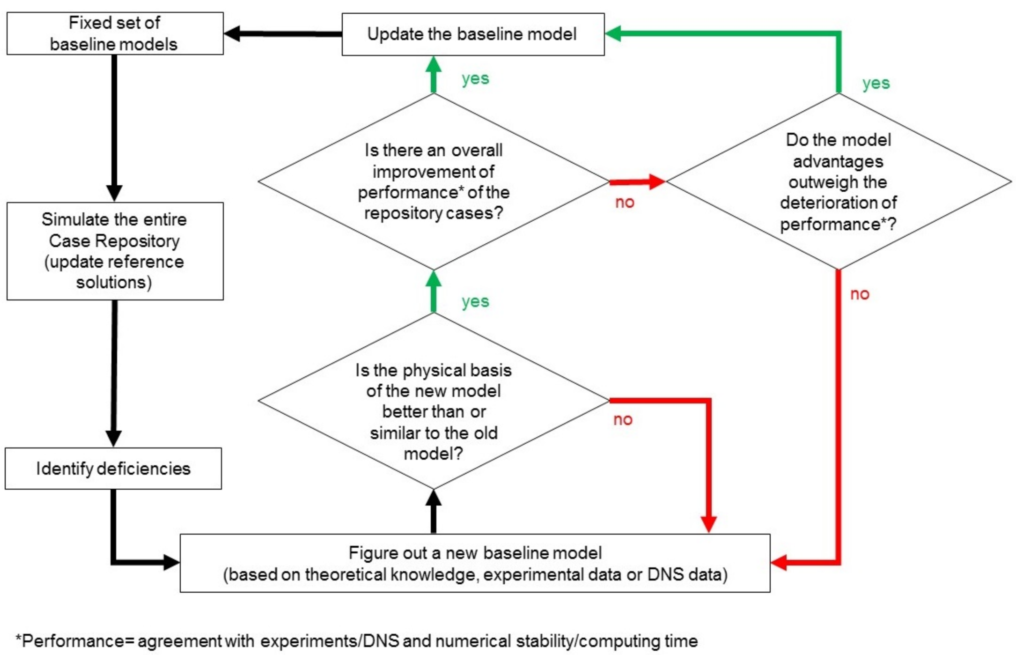


Figure 1: Schematic of the process employed to update baseline models.

2.3. The current collection of cases at HZDR

As described above, the process of testing new models involves the simulation of a large number of validation cases. The case collection at HZDR currently contains 56 pipe and bubble column cases, with the case number continuously growing. Cases differ regarding their geometry, gas injection, bubble sizes, water quality etc. Keywords were introduced to categorise the cases and their sub-cases as illustrated in Table 2. The keywords illustrated are not complete, but rather form a starting point to label existing cases. More cases are to be added in the future in order to diversify the collection and its keywords, as well as to simply increase the amount of validation data.

Case setups are standardised using the best-practice guidelines given in Lucas et al. [45] and simulated using the *HZDR Multiphase Addon for OpenFOAM*, which was customized and extended for the various model developments at HZDR. All cases are under Git version control and managed in GitLab [46]. GitLab Continuous Integration and Continuous Development (CI/CD) mechanisms allow an efficient maintenance of the case collection, as well as its extension by new cases.

Table 2: Overview of cases with all active keywords.

Cases		Geo-metry		Gas in-jection		Gas phases		Bubble size change			Bubble sizes		Water quality			State		Devel-opment	
		Pipe	Bubble column	Bottom	Wall	one-gas	multiple-gas	Fixed	Density-change	Coalescence-breakup	Monodisperse	Polydisperse	Filtered	Tap water	Contaminated	Steady-state	Transient	Fully developed	Developing
Wang et al. (1987)	W1	X		X		X		X			X		X			X		X	
	W2	X		X		X		X			X		X			X		X	
	W3	X		X		X		X			X		X			X		X	
Liu and Bankoff (1993)	LB1	X		X		X		X			X		X			X		X	
	LB2	X		X		X		X			X		X			X		X	
Liu (1998)	L11A	X		X		X		X			X			X		X		X	
	L21B	X		X		X		X			X			X		X		X	
	L21C	X		X		X		X			X			X		X		X	
	L22A	X		X		X		X			X			X		X		X	
Pfleger (1999)			X	X			X			X		X		X			X		X
Deen (2001)			X	X		X		X			X				X		X		X
Hibiki (2001)	H11	X		X		X		X			X		X			X		X	
	H12	X		X		X		X			X		X			X		X	
	H13	X		X		X		X			X		X			X		X	
	H31	X		X		X		X			X		X			X		X	
	H32	X		X		X		X			X		X			X		X	
	H33	X		X		X		X			X		X			X		X	
Shawkat (2008)	S21	X		X		X		X			X		X			X		X	
	S23	X		X		X		X			X		X			X		X	
	S31	X		X		X		X			X		X			X		X	
	S33	X		X		X		X			X		X			X		X	
Hosokawa (2009)	H11	X		X		X		X			X			X		X		X	
	H12	X		X		X		X			X			X		X		X	
	H21	X		X		X		X			X			X		X		X	
	H22	X		X		X		X			X			X		X		X	
Akbar et al. (2012)	13		X	X			X	X				X		X			X		X
	3		X	X		X		X			X			X			X		X
Hosokawa and Tomiyama (2013)	H1	X		X		X		X			X			X		X		X	
	H2	X		X		X		X			X			X		X		X	
	H3	X		X		X		X			X			X		X		X	
	H4	X		X		X		X			X			X		X		X	
Kim (2016)	K1	X		X		X		X			X			X		X		X	
	K2	X		X		X		X			X			X		X		X	
	K3	X		X		X		X			X			X		X		X	
	K4	X		X		X		X			X			X		X		X	
Lucas et al. (2005)	fd_19	X		X		X		X				X	X			X		X	
	fd_37	X		X		X		X				X	X			X		X	
	fd_39	X		X		X		X				X	X			X		X	
	fd_41	X		X		X		X				X	X			X		X	
	fd_59	X		X		X		X				X	X			X		X	
	fd_61	X		X		X		X				X	X			X		X	
	fd_63	X		X		X		X				X	X			X		X	
	fd_85	X		X		X		X				X	X			X		X	
	d_19	X		X		X			X	X		X	X			X			X
	d_37	X		X		X			X	X		X	X			X			X
	d_39	X		X		X			X	X		X	X			X			X
	d_41	X		X		X			X	X		X	X			X			X
	d_59	X		X		X			X	X		X	X			X			X
	d_61	X		X		X			X	X		X	X			X			X
	d_63	X		X		X			X	X		X	X			X			X
	d_85	X		X		X			X	X		X	X			X			X
Lucas et al. (2010)	30	X			X	X				X		X	X			X			X
	40	X			X	X				X		X	X			X			X
	41	X			X	X				X		X	X			X			X
	42	X			X	X				X		X	X			X			X
	52	X			X	X				X		X	X			X			X

2.4. The challenge of evaluating new models

Each case in the case collection produces results that are compared to validation data, i.e. experimental data. Some fully-developed cases produce steady-state results, for other transient cases the results need to be time-averaged. Typical graphs for analyzing results are gas void fraction profiles, liquid and gas velocity profiles, sometimes profiles of turbulent quantities, such as the turbulent kinetic energy. Transient cases produce the respective mean values of the same quantities.

In case a new model is tested the computed results additionally need to be compared with the old baseline reference solution. For each of the plots various curves need to be compared: the computed result using the new model, the reference result obtained with the current baseline model set and the measured data for validation. Examples of such graphs are given in Figure 2. The reason why this kind of evaluation is a time-consuming challenge is twofold:

First, the sheer number of results to evaluate is extensive and hard to handle. It is difficult to demonstrate an overall improvement for the total number of currently 56 cases, each of which produces three to five graphs. Assuming only three graphs for each case, a total of 168 plots has to be evaluated. These plots are typically scattered across case folders. The issue becomes even more pressing with an increasing number of cases and plots, which for baseline development purposes is intended. Therefore, a **workflow** for an automated production of such graphs and their neat presentation in a single report is developed, as will be presented in Section 3.

Second, the information in some graphs can be vague and the decision if there is an improvement of results can become rather subjective. Plotted results need to be evaluated and compared, which is typically done during long and difficult discussions among experts. For a better qualitative judgement and quantification of the performance of new models we will therefore introduce a **fuzzy logic approach** in Section 4 that is meant to aid the decision-making process.

3. Automation of cases in a workflow

Workflow engines are designed to automate the successive execution of commands and applications for large-scale data analyses. The generation of data via workflows follows so-called FAIR principles [47] for a good management of scientific data, which should be transparent and reproducible. Out of a large number of potential workflow systems (such as [48] etc.) for the work presented here we chose Snakemake and developed a workflow, which will be described in the next sections.

3.1. The Snakemake library

Snakemake is a general purpose Python library originally developed for applications in the field of bio-informatics [4]. The library allows to build complex algorithms that process large data-sets using scientific software, and manages the resulting output.

As described by Evdokimov et al. [49] the Snakemake library provides several features that make it a suitable framework for managing the extensive amount of CFD simulations we are interested in here, and their subsequent analysis:

- The syntax of Snakemake offers great flexibility in terms of the scripting languages that jobs can be composed of, which allows individual case-by-case setups.
- The library comes with interfaces to common cluster schedulers required for high-performance computing.
- Snakemake was built for parallization enabling us to spawn parallel case jobs on separate nodes of a cluster to maximize speed. Furthermore, it can adapt the number of jobs in execution to the allocated workflow resources.
- Its modularity allows to rerun downstream analysis and re-process subsets of jobs without the need to rerun the entire pipeline.
- The post-processing output is aggregated and put into a single place with the possibility to include results of hundreds of cases.
- On the low-level a case-by-case approach allows to easily split work among researchers in a team, and

- On the top-level common CI/CD software design practices are applicable and allow the consistent integration of the library into our existing case repository.

3.2. Workflow overview

The details of the workflow on both the case- and the top-level, are described and illustrated in Evdokimov et al. [49]. The following section will briefly summarize the main steps and workflow specifications relevant for our baseline model development.

The workflow consists of the following three steps reflecting a *bottom-up* method of determining job execution:

1. During a *configuration step* an algorithm searches in sub-folders for `Snakefiles` that define the specifications of the pre-processing, solving and post-processing job for each individual case. `Snakefiles` are designed in such a way that they allow the independent workflow execution of a selection of cases. Larger cases that inlude extensive parameter-variations (e.g. Lucas et al. [50] with hundreds of systematic case setup variations) are setup via templates building file structures and sub-cases. Once the *configuration* step has completed, the established file structure is ready for launching the workflow.
2. The workflow then runs the selected cases during the *solution* step. Workflow reports containing the plots of all the cases are produced using the report-generating feature embedded in the Snakemake library.
3. A third optional *post-processing* step produces a case-by-case overview of the agreement of the computed results with the validation data. This step is particularly important for the testing of new baseline models and incorporates the fuzzy logic system described in Section 4. Computed results are not only compared to the validation data, but also to the reference solution obtained by applying the baseline model set.

3.3. Workflow reports

Snakemake reports are generated in the form of static web-pages, which are easily deployed and shared on a web-server. CI/CD mechanisms for the workflow execution allow deploying reports onto web-servers as soon as a certain job has finished making results instantly accessible to the whole research team. A demonstration of the final workflow report via screenshots is presented in Evdokimov et al. [49]. A side panel lists all cases simulated, which upon selection will show the corresponding validation plots on the main page for inspection. For further illustration purposes we refer to the demonstration report provided by the Snakemake library on a public webpage [51]. This report gives an impression of the typical report structure for results.

The Snakemake workflow gives us an efficient tool to produce a large amount of plots and to represent them in a well-structured report. The evaluation of these reports for the baseline model development remains the task of the user, but can be supported by artificial intelligence tools, such as the fuzzy logic controller described in the next section.

4. Evaluation of results

The decision whether or not a computed *simulation* result is better than the *baseline* reference solution can be quite easy to make, such as in the plot illustrated on the left hand side of Figure 2, or it can be less obvious, such as the example on the right hand side.

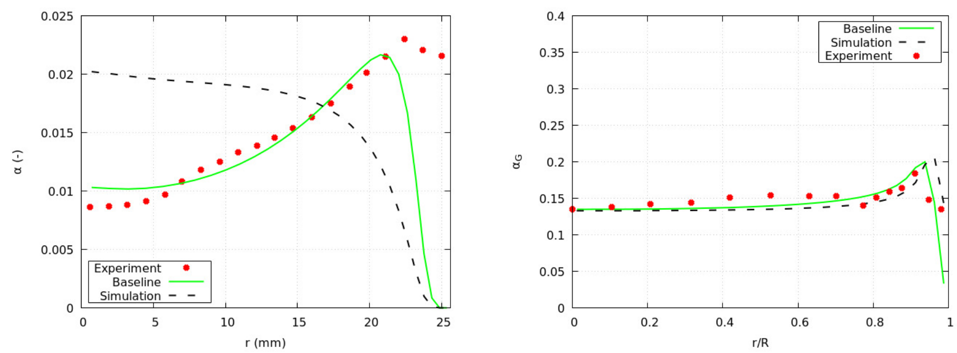


Figure 2: Gas void fraction profiles of fullyDeveloped_39 (left) and case L11A (right).

Fuzzy logic is a subset of artificial intelligence, which can accommodate the imprecision of the real world and support such a decision-making process. By using linguistic variables fuzzy logic can deal with imprecise, vague and uncertain information and approximate human reasoning.

The `scikit-fuzzy` library ([52], [53]) allows the integration of such fuzzy logic tools into our workflow, where it can be utilized for the testing of new models. From the system design point of view the library takes responsibility for the correct and robust implementation of the underlying algebra and the numerical algorithms, while on the top level we investigate metrics and their value boundaries suitable to our multiphase CFD problems. In essence, we propose a *fuzzy logic controller* for the purpose of evaluating and comparing CFD results. In the next sections we briefly describe how this controller works.

4.1. The fuzzy logic system

Fuzzy logic uses a specific terminology, which is briefly introduced here for a better understanding of the following sections. When a value is referred to as being crisp this means it is explicit and concise. *Fuzzification* is the process of decomposing such crisp values into a spectrum of different linguistic categories, referred to as fuzzy sets. A typical fuzzy logic system is illustrated in Figure 3. The system consumes and *fuzzifies* several crisp inputs, which means numerical floating-point or integer values are *mapped* onto linguistic sets via *membership functions*. These *membership functions* are normalized and might be considered as weight-factors of the expert system. *Inference logic* combines the numerous *fuzzy input sets* via a set of rules to produce *fuzzy output sets*, which themselves represent linguistic sets. Finally, these output sets are summed up, *defuzzified* and mapped onto a single crisp output value.

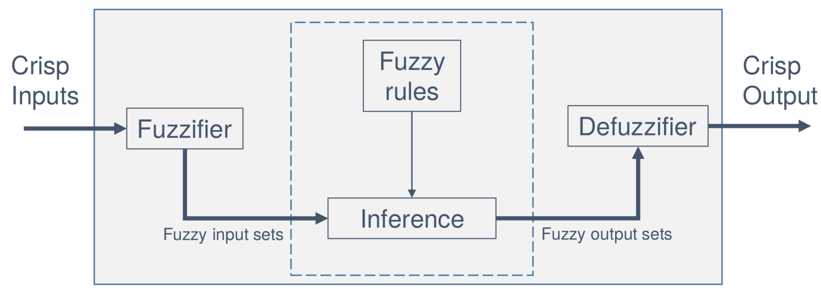


Figure 3: Schematic of a fuzzy logic system.

The inputs specified for the fuzzy system demonstrated here are two error metrics that will be described in the next section. One of the advantages of the fuzzy logic approach lies in its modularity allowing us to extend this list of fuzzy input variables in the future. In our case the desired output of the fuzzy logic system is a value between 0 and 1 quantifying how good the computed CFD result agrees with the measured validation data. We will refer to this crisp output as **goodness-value** G of the CFD prediction.

4.2. The crisp inputs: error metrics

The crisp inputs for our fuzzy logic system are specific error metrics that describe the similarity of two curves: our computed result, such as the gas void fraction profile, and the corresponding experimental data for validation. Various metrics could be used, but for demonstration purposes the number here is limited to two, which complement each other. The error metrics below are defined using the example of evaluating sorted sets of gas void fraction data. Hereby, the sorted set represents the radial profile in case of round geometries or the lateral profile in case of rectangular columns. Metrics for the other validation fields are computed accordingly.

As a quantitative measure the **Mean relative error MRE** is defined as:

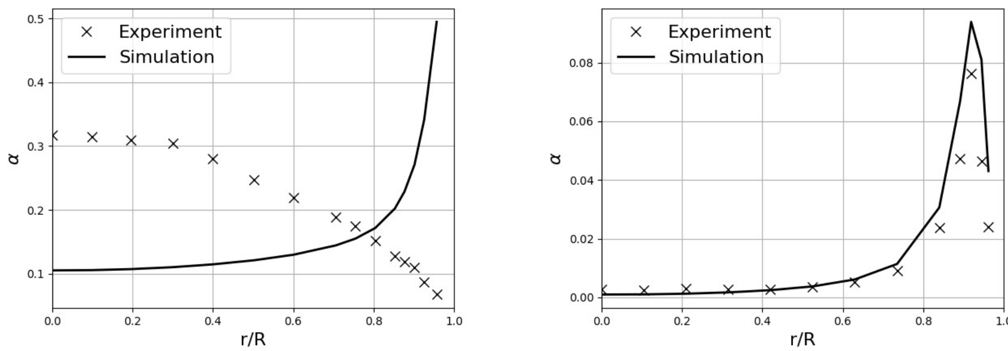
$$\mathrm{MRE} = \frac{1}{N}\sum_{n=1}^{N}\frac{|\alpha_{n,sim} - \alpha_{n,exp}|}{\bar{\alpha}_{exp}} \tag{1}$$

with N being the number of values, the computed values of the gas void fraction α_{sim} and the experimentally measured ones α_{exp}, and $\bar{\alpha}$ the arithmetic average. The MRE describes the quantitative distance between two curves along the ordinate. It is normalized by the mean experimental value for the entire profile in order to avoid a heavier penalty of discrepancies for data points with smaller experimental values. Possible values for the MRE range from 0 to ∞. We therefore limit values to a maximum of 1. Note that a case with a MRE > 1 therefore will not receive a heavier penalty than cases with a MRE $= 1$. Most cases in our collection, however, fall into the range of $0 <$ MRE < 1.

As a qualitative measure the **Pearson correlation coefficient PCC** is defined as the covariance of the two gas void fraction data sets divided by the product of their standard deviation:

$$\mathrm{PCC} = \frac{\sum_{n=1}^{N}(\alpha_{n,sim} - \bar{\alpha}_{sim})(\alpha_{n,exp} - \bar{\alpha}_{exp})}{\sqrt{\sum_{n=1}^{N}(\alpha_{n,sim} - \bar{\alpha}_{sim})^2}\sqrt{\sum_{n=1}^{N}(\alpha_{n,exp} - \bar{\alpha}_{exp})^2}} \tag{2}$$

The Pearson correlation coefficient determines the similarity of the two data sets, i.e. the shapes of the two curves. The range of the Pearson coefficient is $-1 <$ PCC < 1. Examples of PCC values are illustrated in Figure 4, where simulation profiles are compared with experimental data. A negative coefficient indicates a negative linear correlation and a low similarity between curves, such as in the case of a predicted wall peak for a measured core peak profile. A positive coefficient represents a positive linear correlation of the data set meaning a high similarity of shapes, as illustrated for the almost matching profiles on the right hand side.



PCC= -0.83 PCC= 0.98

Figure 4: Pearson correlation coefficients computed for different data sets of gas void fraction profiles.

The PCC criteria is not sensitive to shifts of the profiles along the ordinate, but sensitive to shifts along the abscissa. Thus, it complements the MRE metric, which is mainly sensitive to shifts along the ordinate.

4.3. Membership functions

A fuzzy set of linguistic terms is assigned to each input and output metric as illustrated in Table 3. The process of turning crisp variables into linguistic sets is termed *fuzzification*.

Table 3: Linguistic sets assigned to each variable.

Variable	Linguistic sets	Parameters a,b,c
MRE	”low”	0, 0, 0.3
	”medium”	0, 0.3, 1.0
	”high”	0.3, 1.0, 1.0
PCC	”high”	0.4, 1.0, 1.0
	”medium”	-1.0, 0.4, 1.0
	”low”	-1.0, -1.0, 0.4
G	”perfect”	0.7, 1.0, 1.0
	”good”	0.5, 0.7, 0.9
	”tolerable”	0.3, 0.5, 0.7
	”bad”	0.1, 0.3, 0.5
	”defect”	0.0, 0.0, 0.3

Membership functions $\mu_i(x)$ tie the input and output variables x for each linguistic set item i of the above table to a number, which can be understood as an impact value. Piece-wise *triangle* functions are used:

$$\mu_i(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{cases} \tag{3}$$

with parameters a, b and c specified separately for each item of the set as listed in Table 3. The triangle-shaped membership functions assigned to the linguistic terms listed in Table 3 are illustrated in Figure 5.

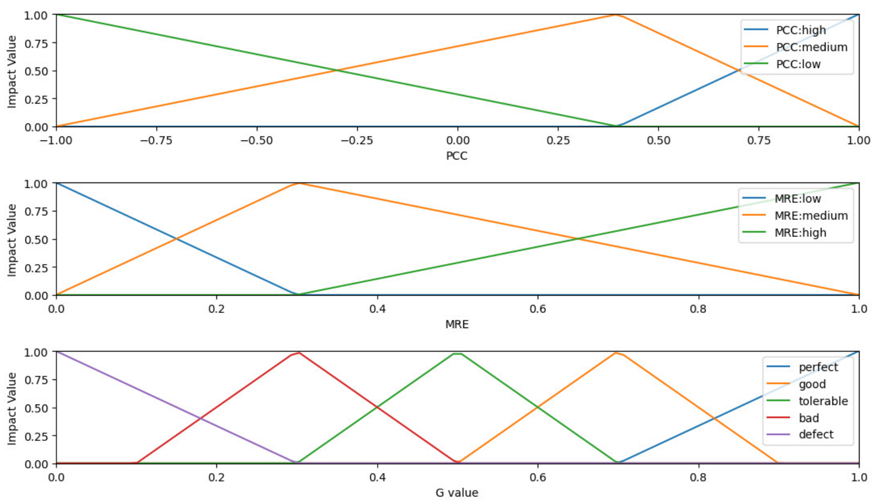


Figure 5: Membership functions for the fuzzy input and output sets. The x axis represents the crisp value space, the y value represents the membership function output value for each linguistic term.

For a specific metric value the membership function output represents a fuzzy degree of membership in the qualifying linguistic set. Thus, value intervals are constrained to a uniform range from zero to one for all input and output metrics, and the fuzzy set is *normalized*. For a crisp value x a fuzzy set now is defined as a pair:

$$A(x) = (i, \mu_i(x)) \tag{4}$$

where item i is a linguistic term and $\mu_i(x)$ the membership function expressing the degree of membership in that term.

4.4. Fuzzy rule base

Operations can be defined on the various fuzzy sets by means of their membership function. This allows us to prescribe an intuitive rule base, which is easy to interpret.

The current workflow evaluation consists of the following five rules:

1. `If PCC:low AND MRE:high` $\rightarrow$ `Defect G`
2. `If (PCC:low AND MRE:medium) OR (PCC:medium AND MRE:high)` $\rightarrow$ `Bad G`
3. `If (PCC:medium AND MRE:medium)`
   `OR (PCC:high AND MRE:high)`
   `OR (PCC:low AND MRE:low)` $\rightarrow$ `Tolerable G`
4. `If (PCC:high AND MRE:medium) OR (PCC:medium AND MRE:low)` $\rightarrow$ `Good G`
5. `If PCC:high AND MRE:low` $\rightarrow$ `Perfect G`

A simplified visual representation for the target "Goodness" metric G is illustrated in Figure 6.

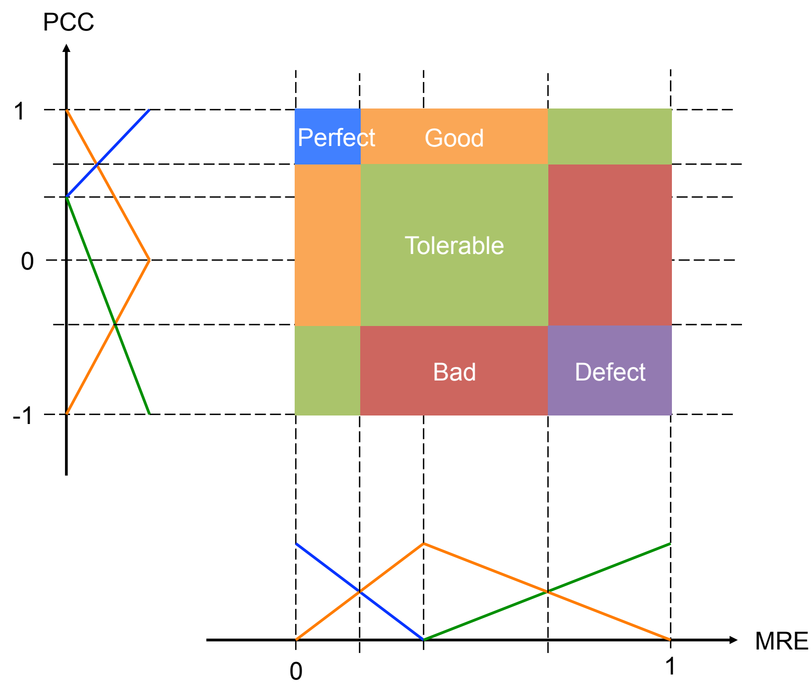


Figure 6: A schematic view of the relationship between the fuzzy input metrics and the output metric G.

Fuzzy operators are applied to obtain one value representing the result of each rule. Logical min/max operators correspond to the AND/OR rules mentioned above [54]. Input for these operators are the membership values of the fuzzified input variables, and the output is a single value.

An example is illustrated in Figure 7 for a case with $PCC = 0.49$ and $MRE = 0.36$. A single fuzzy rule, rule 4 for the fuzzy set *Good* of the G metric, is exemplified. The condition `(PCC:high AND MRE:medium) OR (PCC:medium AND MRE:low)` can be translated into `max(min(0.15, 0.91), min(0.85, 0)) = 0.15`. This output is used for the implication of the fuzzy rule as described in the next section.

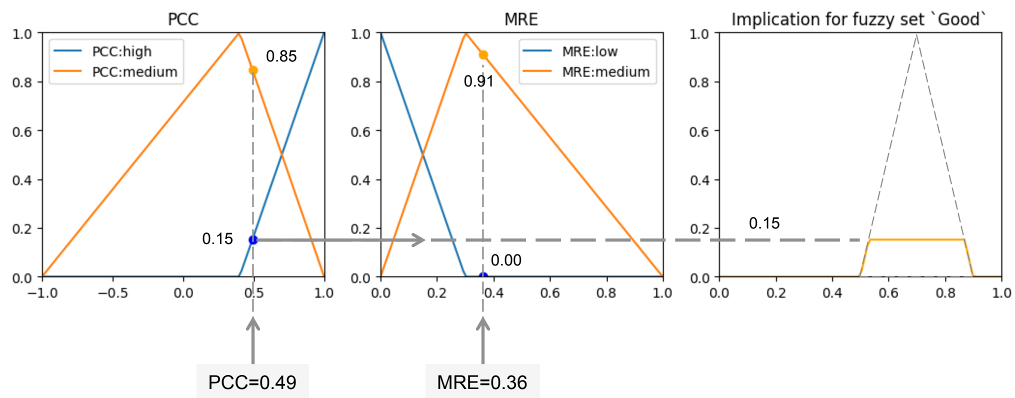


Figure 7: Application of rule 4 via fuzzy operators for an example of two input metrics.

4.5. Implication and defuzzification

An implication method turns the single output value into the consequence of the fuzzy rule. The consequent fuzzy sets are represented by the membership functions of the output sets. Input for the implication method is the value outputs given by each of the antecedent rules, such as rule 4 above.

Mamdani-type inference was applied for the implication process as it is an intuitive method, well-suited to human input, allowing an interpretable rule base and with widespread acceptance [54]. It works with the *min* implication operator to infer the output functions, as it is illustrated for rule 4 on the right hand side of Figure 7. The five fuzzy rules produce the output functions for the five potential sets of the goodness variable. Figure 8 illustrates the whole inference process for the above example with $PCC = 0.49$ and $MRE = 0.36$. In the example only three out of five output sets qualify during the inference step.

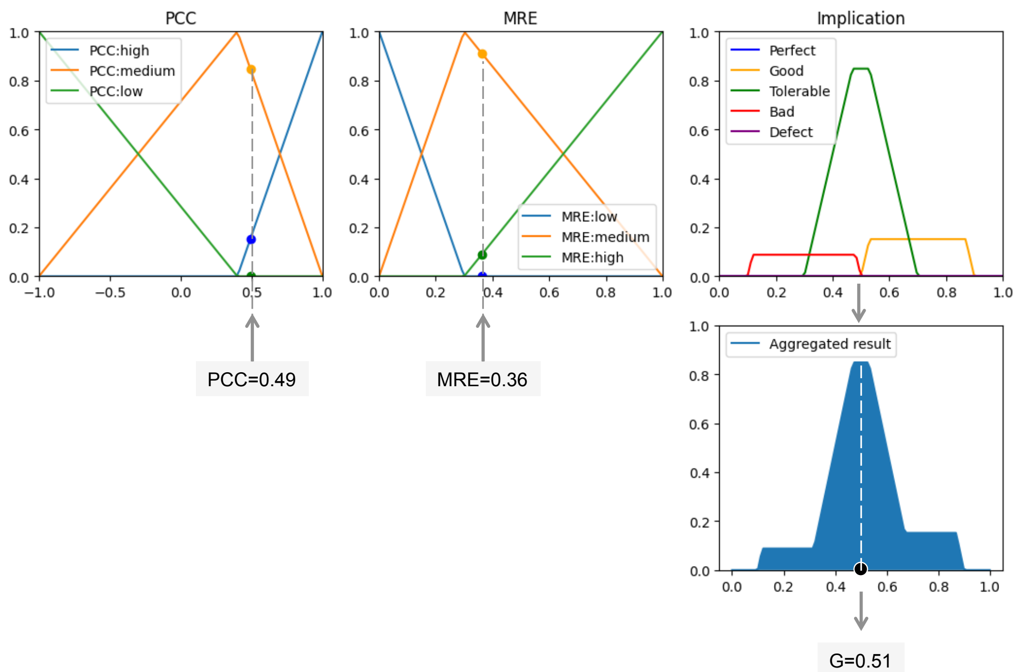


Figure 8: Inference system including all rules for an example of two input metrics.

The consequent functions of all rules then need to be aggregated, which is done by a simple *max* function. In a last step the aggregated function is defuzzified. The `scikit-fuzzy` library allows to choose from several methods suggesting different trade-offs between the smoothness of the response surface and the robustness of the final defuzzification step. In the model evaluation attempted here we tried to track the smallest variations to allow a continuous differentiation of results. Therefore, our chosen defuzzification method relies on area measurements of the aggregated result. Defuzzification is performed by the *bisector* method, which finds the vertical line that divides the aggregated result into two sub-regions of equal area. The response surface of our current two-input, one-output, five-rule fuzzy logic system is illustrated in Figure 9.

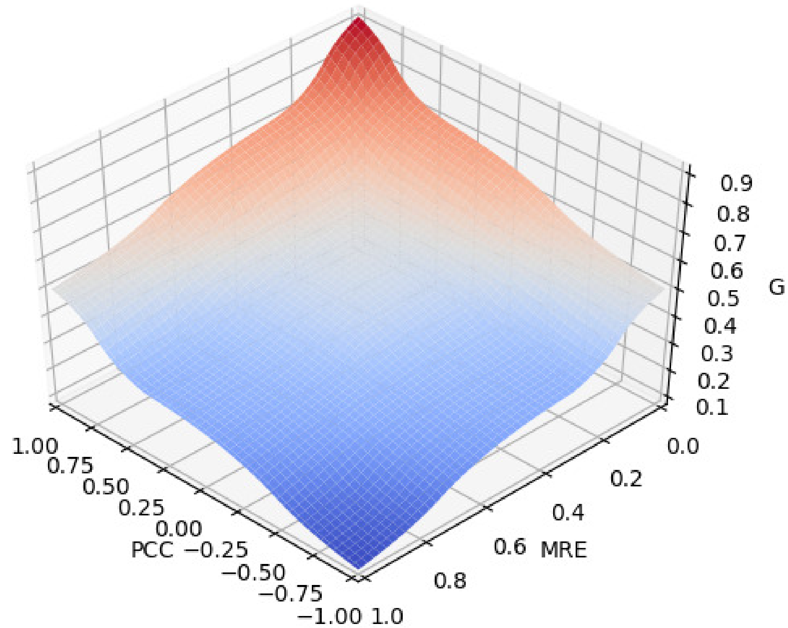


Figure 9: Response surface of the fuzzy logic system.

4.6. Evaluation for the case collection

The fuzzy logic approach offers a convenient framework for an evaluation algorithm, which could be easily repurposed, extended and redefined. The alternative usage of an equivalent multidimensional target function requires the strict definition of potentially controversial input values fitting a specific output metric. Fuzzy logic makes it easy to reach a consensus amongst researchers by using relations between linguistic sets derived from these values. Errors divided into a *low*, *medium* and *high* subset are easily agreed on, but certain values fitting these sets are not.

As described above for each case in the case collection several fields are investigated for the validation of the CFD results, e.g. gas void fractions or velocity profiles. A goodness value G is computed for each of those individual validation fields and is meant to describe how good the predicted profile agrees with the experimental data.

The average over all these individual goodness values for the available validation fields produces a single value for each case. This single value represents the overall goodness of the CFD prediction for the case investigated. In the future different weightings for the various validation fields could be considered.

When testing a new model the above procedure is done twice, for the results using the new model and for the reference solution obtained with the baseline model set. An improvement or deterioration of the CFD results for a specific case can then be expressed via the difference in the overall goodness of the computed and the baseline result.

5. Demonstration case: Testing a new lift force correlation

To demonstrate the efficiency of the baseline workflow and its fuzzy logic tool, this section will now investigate results when testing a new lift force correlation. We will refer to the results obtained with the Tomiyama correlation [17] as *baseline* results, and the results with a new correlation for the lift coefficient proposed by Hessenkemper et al. [55] as *computed* results. This section does not aim to validate the new correlation, but to demonstrate the benefits and future potential of the developed workflow tools in assisting the baseline model development.

We will give an impression of the performance and reporting features of the workflow when analysing a certain selection of cases. The evaluation of results via fuzzy logic will then be verified for a selected case. Finally, an overview of the overall goodness evaluation for all cases is presented.

5.1. Workflow performance

For the following analysis we restrict our selection of respository cases to the ones that have already been published elsewhere, as listed in Table 4. The individual case setups in OpenFOAM are described in the references given, and are to be found in the *HZDR Multiphase Case Collection for OpenFOAM* [56]. Including all sub-cases the total number of cases investigated here is 36. The simulation of each individual sub-case demands three to eight processors and an execution time of up to two hours.

Table 4: List of cases for the workflow demonstration.

Experimental reference	Cases	Setup	CPUs/Case
Hosokawa and Tomiyama [57]	4	[58]	4
Liu [59]	4	[58], [60]	4
Shawkat et al. [61]	4	[60]	4
Hosokawa and Tomiyama [62]	4	[60], [63]	8
Kim et al. [64]	4	[63]	8
Lucas et al. [50]	16	[65]	3

The Snakemake workflow processes all these cases with a total execution time of ∼5h (the entire case collection takes ∼10h). A total of 236 plots (baseline relevant and auxiliary) is generated automatically. In the plots relevant for the baseline evaluation the computed results are compared to both the experimental data and the reference solution. Examples of such plots are illustrated in Figure 2. The fuzzy logic evaluation of these results during the optional post-processing step is done with an execution time of only four minutes.

5.2. Verification of the fuzzy logic output

For a single case this section will now analyse the computed results and compare them with the error metrics and the goodness output computed by the fuzzy logic system. Figure 10 illustrates the validation plots produced for exemplary case L11A of Liu [59], and associated values of error metrics and the G output metric.

The predicted gas void fraction profiles show a generally good shape, both the baseline and the computed profile. This generally good agreement with the experiment is well represented by high values of the PCC. The result with the Hessenkemper lift appears slightly deteriorated due to a shift of the wall peak towards the wall, which is well captured by a drop of the respective PCC. There are no significant changes in MRE. The drop in overall goodness and a negative ΔG reflect the subjective evaluation of the gas void fraction profiles.

The profiles for the turbulent kinetic energy are examples of curves with distinctively different shapes compared to the experimental data (wall-peak vs. core-peak). This judgement is well captured by very low values for the PCC, as well as very high values for the MRE. Comparing the baseline result to the new model no significant changes are observed. Consequently, the ΔG computed by the fuzzy logic system is relatively small.

The profiles for the liquid velocity show a good agreement with the experiment in terms of the shape. The difference to the experimental data slightly increases with the new model, as correctly captured by the MRE metric.

All the resulting G values of the validation fields are averaged to produce one value for the overall goodness of the CFD predictions for the case. The difference in results between the two lift force models is expressed by the difference ΔG. For the L11A case the new lift model slightly decreases the overall G and deteriorates results, which corresponds to our judgement for the plots in Figure 10.

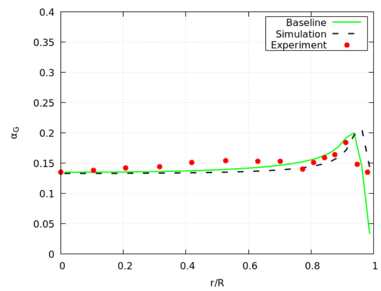
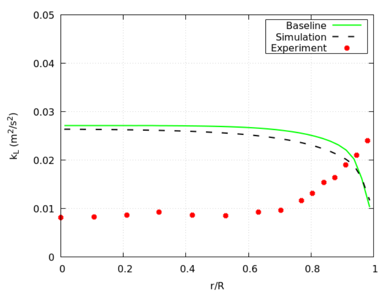
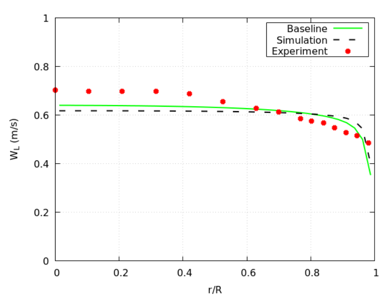
Validation plots	Metrics	Baseline	Simulation	ΔG
	MRE PCC **G**	0.089 0.701 **0.713**	0.091 0.349 **0.651**	**-0.062**
	MRE PCC **G**	1.064 -0.933 **0.095**	0.962 -0.973 **0.099**	**0.004**
	MRE PCC **G**	0.066 0.793 **0.770**	0.082 0.683 **0.713**	**-0.057**
	Overall G	**0.526**	**0.487**	**-0.039**

Figure 10: Goodness evaluation for the validation fields of case L11A.

5.3. Overview of all case evaluations

The above evaluation of results expressed via the G metric is performed for each of the cases under investigation. Figure 11 illustrates the overall goodness values of all cases, represented as bars ranging from zero to one. Numbers next to each case indicate in green the cases which have improved, and in red the deteriorated ones, such as L11A on the top. A ΔG value averaged over all cases aims to quantify the general trend in results. Thus, the plot allows to obtain a first overview of the general performance of a new model.

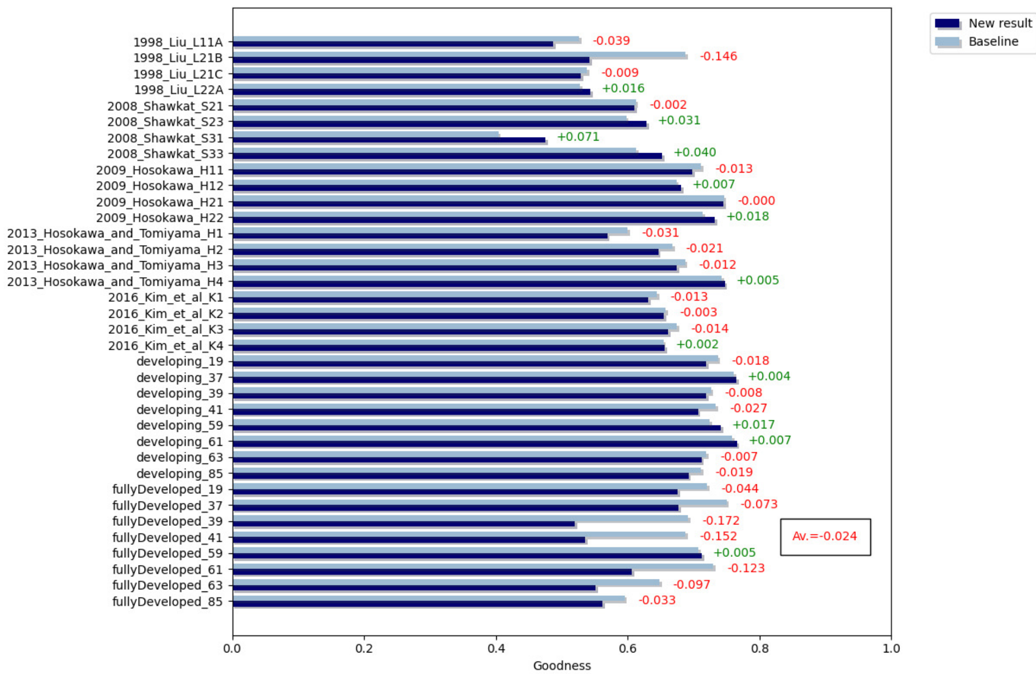


Figure 11: Plots for the overall goodness of all 36 demonstration cases. Light blue bars represent the reference solution obtained with the baseline lift model, deep blue bars the solution obtained with the new lift model. Numbers next to each case indicate the difference in overall goodness, with red indicating a deterioration and green an improvement of results.

Similar plots are produced for the individual validation fields, such as the gas void fraction plot presented in Figure 12. The goodness values are based on the computed mean relative errors and Pearson correlation coefficients illustrated in Figure 13.

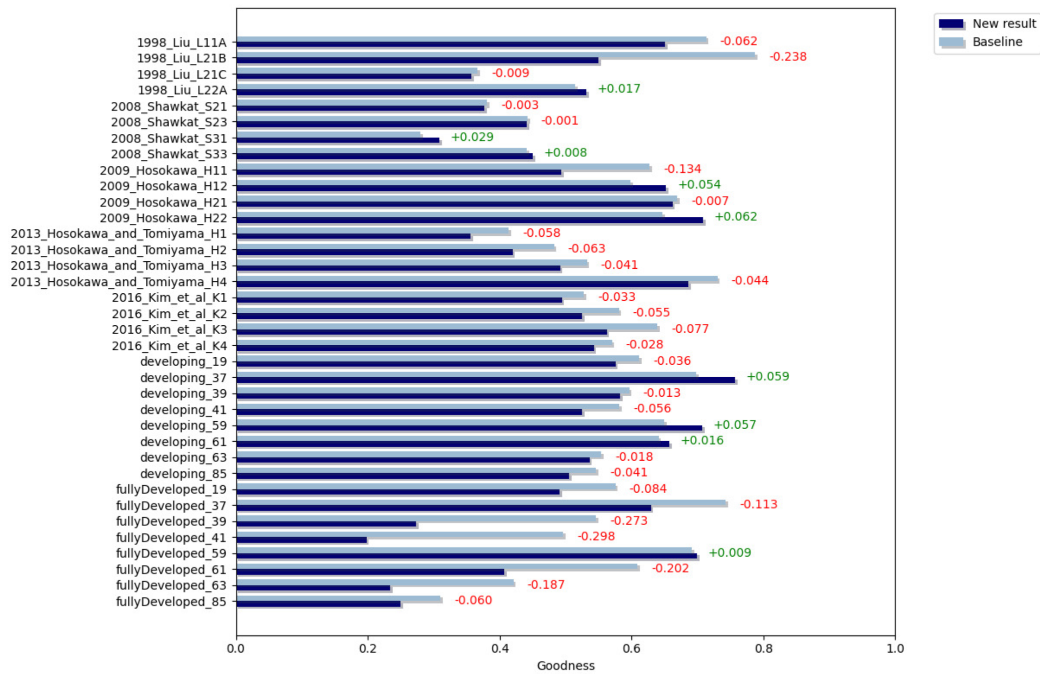


Figure 12: Goodness output for the gas void fraction prediction of all 36 demonstration cases.

Plots visualizing the change in goodness for each case, such as the ones above, have proven to be a useful tool for analysing the performance of a new model. Not only do they provide a neat case-by-case overview of results, but also help pointing the researcher to the cases most affected by a new model. Figure 12 indicates a slight change of gas void fraction results for case L11A, but a more substantial change for case fullyDeveloped_39. This is confirmed by the corresponding plots in Figure 2. Table 5 illustrates the detailed input and output metrics for both cases, with the quality of CFD predictions adequately quantified by the fuzzy logic system.

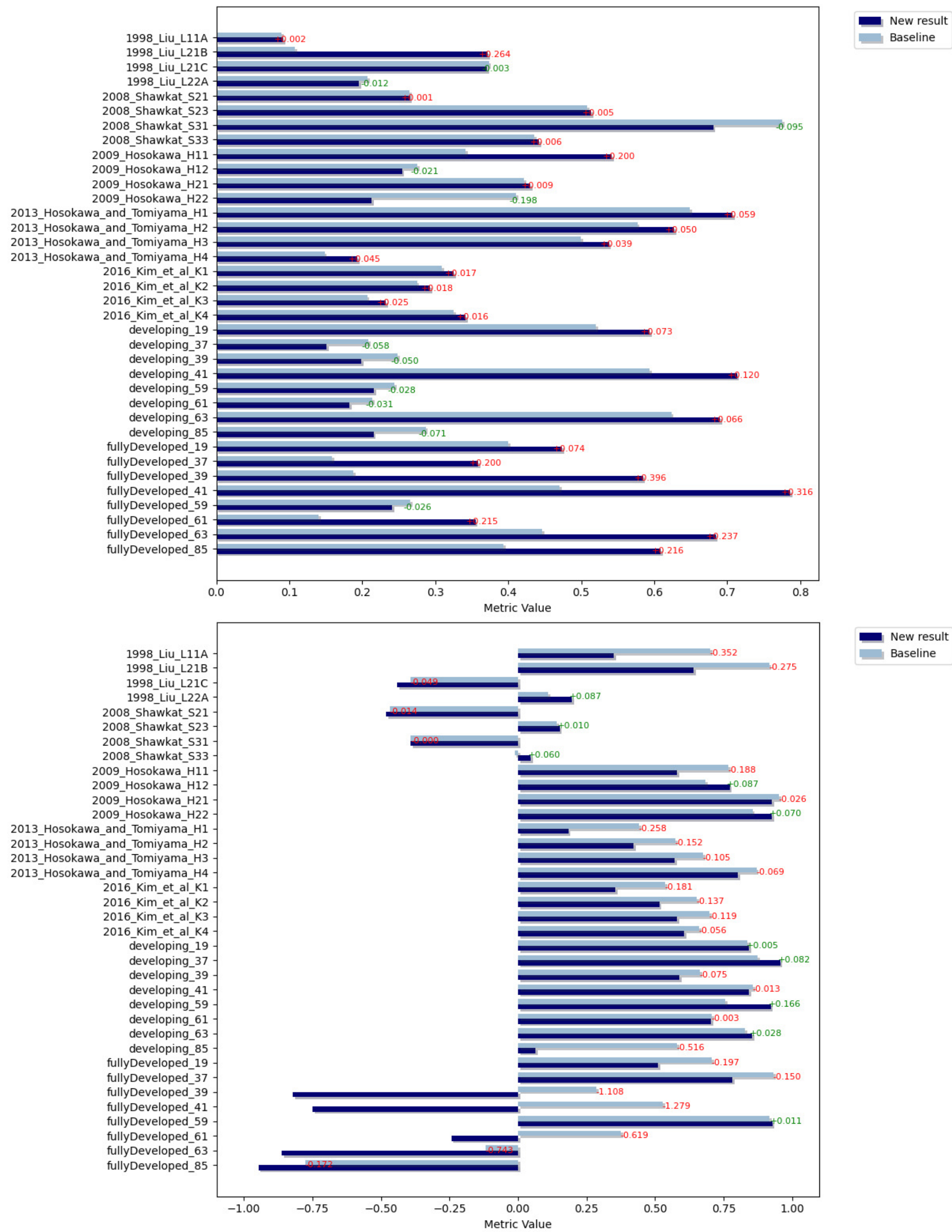
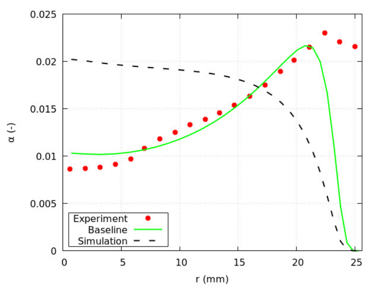
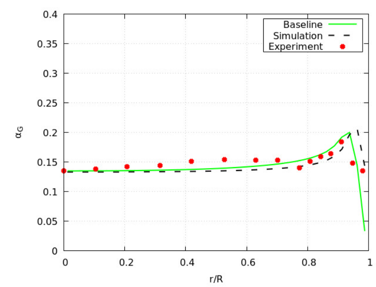


Figure 13: Computed error metrics for the gas void fraction prediction of all 36 demonstration cases. Top: Mean relative error. Bottom: Pearson correlation coefficient.

Table 5: Comparison of the computed fuzzy logic metrics for the gas void fraction profiles in Figure 2.

	fd_39	**L11A**
		
Baseline		
MRE	0.188	0.089
PCC	0.287	0.701
G	**0.546**	**0.713**
New model		
MRE	0.585	0.091
PCC	-0.822	0.349
G	**0.273**	**0.651**
Δ G	**-0.273**	**-0.062**

The presented fuzzy logic plots above are a powerful tool assisting in the model evaluation and comparison for the baseline model development using a large number of cases. Note, however, that results will not cease to require the careful interpretation by the experienced researcher taking into account the underlying physical phenomena. An obvious limitation of the current validation strategy is that bubbly flow cases typically require a set of interdependent closures rather than isolated, individual models. This means the computed error metrics and corresponding goodness evaluation is only able to demonstrate the accuracy of all closure models combined. How well each individual closure model performs remains a task of careful interpretation by the investigator.

For a more detailed analysis of a new closure model the researcher can facilitate the keywords presented in Table 2, which are assigned to each sub-case. Via keyword selection during the configuration step of the workflow the model analysis can be tailored to specific cases of interest with certain features.

The fuzzy logic analysis demonstrated here constitutes a first step towards a more automated model evaluation. In the future a more refined control of the results is possible by introducing alternative and/or additional metrics. Furthermore, core and near-wall regions could be analysed separately adopting different error metrics and fuzzy sets for these individual flow regions if needed. The optimal use of fuzzy sets and metric combinations will be determined by the expert community and will evolve by gaining experience over time.

6. Conclusions

The development of closure models that can predict the behaviour of multiphase flows continues to be a challenge due to the increasing number of available model options, and their intricately-linked interactions. The baseline strategy offers a way towards a more sustainable model development. By setting the goal of establishing a fixed set of models and related model constants it is turning away from case-specific tuning, and instead gives priority to the generality and predictive capabilities of models. The practical implementation of such a strategy involves the continuous validation of a large number of cases available in the literature, which continues to grow. Within the last few years of baseline research the HZDR team has accumulated a diverse set of such validation cases and built an extensive case collection.

We have presented here a Snakemake workflow that now allows the flexible and fully-automated pre-processing, simulation and post-processing of all these OpenFOAM cases at scale. The automated production of hundreds of validation plots and their representation in a single, easily accessible web-report allow an efficient analysis of our CFD results and their comparison to the experimental validation data.

Such advanced case management enables us to integrate artificial intelligence tools that assist in the analysis and evaluation of the extensive amount of results. The work presented here introduces a fuzzy logic controller quantifying the agreement of computed CFD results with the validation data. By comparing new results with a reference solution obtained with the baseline model set this fuzzy logic tool allows the systematic evaluation of new Euler-Euler models. The testing of a new lift force correlation demonstrates the potential of such an approach with comprehensive plots illustrating neat case-by-case comparisons of the accuracy of the CFD predictions.

The further development of the fuzzy logic evaluation and its extension by more input metrics is ongoing. A particular focus will be on metrics describing the convergence behaviour and runtime performance of cases to obtain an alternative criteria by which to compare various Euler-Euler models. Furthermore, the uncertainty of the validation data should be included into the evaluation. The application of fuzzy logic is just one of many potential ways of including data science tools into the baseline model development. Another valuable area of further work is the integration of case-specific keywords into the model evaluation for further categorisation, which suits various machine learning algorithms.

Acknowledgements

It is a pleasure to acknowledge the helpful discussions with Dr. rer. nat. P. Steinbach from the Department of Computational Sciences at HZDR regarding the Snakemake library. This work was partly supported by the "Helmholtz European Partnering" program in the project "Crossing borders and scales - an interdisciplinary approach (CROSSING)".

References

[1] R. Sugrue, B. Magolan, N. Lubchenko, E. Baglietto, Assessment of a simplified set of momentum closure relations for low volume fraction regimes in STAR-CCM+ and OpenFOAM, Annals of Nuclear Energy 110 (2017) 79 – 87.

[2] G. Besagni, F. Inzoli, T. Ziegenhein, Two-phase bubble columns: A comprehensive review, ChemEngineering 2 (2018).

[3] M. Colombo, R. Rzehak, M. Fairweather, Y. Liao, D. Lucas, Benchmarking of CFD modelling closures for two-phase turbulent bubbly flows, in: NURETH-18.

[4] J. Köster, S. Rahmann, Snakemake - a scalable bioinformatics workflow engine, Bioinformatics 28 (2012) 2520–2522.

[5] M. Cornwell, et al., VIPER: Visualization Pipeline for RNA-seq, a Snakemake workflow for efficient and complete RNA-seq analysis, BMC Bioinformatics 19 (2018).

[6] D. Lucas, R. Rzehak, E. Krepper, T. Ziegenhein, Y. Liao, S. Kriebitzsch, P. Apanasevich, A strategy for the qualification of multi-fluid approaches for nuclear reactor safety, Nuclear Engineering and Design 299 (2016) 2–11. CFD4NRS-5.

[7] R. Rzehak, T. Ziegenhein, S. Kriebitzsch, E. Krepper, D. Lucas, Unified modeling of bubbly flows in pipes, bubble columns, and airlift columns, Chemical Engineering Science 157 (2017) 147–158.

[8] Y. Liao, T. Ma, E. Krepper, D. Lucas, J. Fröhlich, Application of a novel model for bubble-induced turbulence to bubbly flows in containers and vertical pipes, Chemical Engineering Science 202 (2019) 55–69.

[9] Y. Liao, R. Rzehak, D. Lucas, E. Krepper, Baseline closure model for dispersed bubbly flow: Bubble coalescence and breakup, Chemical Engineering Science 122 (2015) 336 – 349.

[10] Openfoam-dev Github, `https://github.com/OpenFOAM/OpenFOAM-dev`, 2020.

[11] F. Schlegel, M. Draw, I. Evdokimov, S. Hänsch, H. Khan, R. Lehnigk, R. Meller, G. Petelin, M. Tekavčič, HZDR Multiphase Addon for OpenFOAM, `https://rodare.hzdr.de/record/768`, 2021.

[12] M. Ishii, N. Zuber, Drag coefficient and relative velocity in bubbly, droplet or particulate flows, AIChE Journal 25 (1979) 843–855.

[13] L. Schiller, Z. Naumann, A drag coefficient correlation, Zeitschrift des Vereins Deutscher Ingenieure 77 (1935) 318–320.

[14] A. Tomiyama, I. Kataoka, T. Sakaguchi, Drag coefficients of bubbles: 1st Report, Drag coefficients of a single bubble in a stagnant liquid, Transactions of the Japan Society of Mechanical Engineers Series B 61 (1995) 2357–2364.

[15] A. Tomiyama, G. Celata, S. Hosokawa, S. Yoshida, Terminal velocity of single bubbles in surface tension force dominant regime, International Journal of Multiphase Flow 28 (2002) 1497–1519.

[16] A. Tomiyama, I. Kataoka, I. Zun, T. Sakaguchi, Drag coefficients of single bubbles under normal and micro gravity conditions, JSME International Journal Series B Fluids and Thermal Engineering 41 (1998) 472–479.

[17] A. Tomiyama, H. Tamai, I. Zun, S. Hosokawa, Transverse migration of single bubbles in simple shear flows, Chemical Engineering Science 57 (2002) 1849 – 1858.

[18] D. Legendre, J. Magnaudet, The lift force on a spherical bubble in a viscous linear shear flow, Journal of Fluid Mechanics 368 (1998) 81–126.

[19] F. J. Moraga, F. J. Bonetto, R. Lahey, Lateral forces on spheres in turbulent uniform shear flow, International Journal of Multiphase Flow 25 (1999) 1321–1372.

[20] A. Burns, T. Frank, I. Hamill, J.-M. Shi, The Favre averaged drag model for turbulent dispersion in Eulerian multi-phase flows, in: 5th International Conference on Multiphase Flow.

[21] A. D. Gosman, C. Lekakou, S. Politis, R. I. Issa, M. K. Looney, Multidimensional modeling of turbulent two-phase flows in stirred vessels, AIChE Journal 38 (1992) 1946–1956.

[22] M. Lopez de Bertodano, Turbulent bubbly two-phase flow in a triangular duct, Ph.D. thesis, Rensselaer Polytechnic Institution, 1992.

[23] C. Crowe, J. Schwarzkopf, M. M. Sommerfeld, Y. Tsuji, Multiphase Flows with Droplets and Particles, 2011.

[24] H. Lamb, Hydrodynamics, 1993.

[25] S. Hosokawa, A. Tomiyama, S. Misaki, T. Hamada, Lateral migration of single bubbles due to the presence of wall., in: ASME 2002 Joint US-European Fluids Engineering Division Conference.

[26] S. P. Antal, R. Lahey Jr, J. Flaherty, Analysis of phase distribution in fully developed laminar bubbly two-phase flow, International Journal of Multiphase Flow 17 (1991) 635–652.

[27] T. Frank, Advances in computational fluid dynamics (CFD) of 3-dimensional gas-liquid multiphase flows, in: NAFEMS Seminar: Simulation of Complex Flows (CFD)-Applications and Trends, pp. 1–18.

[28] A. Tomiyama, Struggle with computational bubble dynamics, Multiphase Science and Technology 10 (1998) 369–405.

[29] F. Menter, M. Kuntz, R. Langtry, Ten years of industrial experience with the SST turbulence model, Turbulence, Heat and Mass Transfer 4 (2003) 625 – 632.

[30] B. Launder, D. Spalding, The numerical computation of turbulent flows, Computer methods in applied mechanics and engineering 3 (1974) 269–289.

[31] D. Wilcox, Turbulence modeling for CFD, volume 2, 1998.

[32] T. Ma, C. Santarelli, T. Ziegenhein, D. Lucas, J. Fröhlich, Direct numerical simulation-based Reynolds-averaged closure for bubble-induced turbulence, Physical Review Fluids 2 (2017).

[33] R. Rzehak, E. Krepper, CFD modeling of bubble-induced turbulence, International Journal of Multiphase Flow 55 (2013) 138–155.

[34] Y. Sato, M. Sadatomi, Momentum and heat transfer in two-phase bubble flow - I, Theory, International Journal of Multiphase FLow 7 (1981) 167–177.

[35] Y. Liao, R. Oertel, S. Kriebitzsch, F. Schlegel, D. Lucas, A discrete population balance equation for binary breakage, Int J Numer Meth Fluids (2018) 1–14.

[36] M. Ishii, S. Kim, J. Kelly, Development of interfacial area transport equation., Nuclear Engineering and Technology 37 (2005) 525–536.

[37] C. Coulaloglou, L. Tavlarides, Description of interaction processes in agitated liquid-liquid dispersions, Chemical Engineering Science 32 (1977) 1289–1297.

[38] F. Lehr, M. Millies, D. Mewes, Bubble-size distributions and flow fields in bubble columns, AIChE Journal 48 (2002) 2426–2443.

[39] H. Luo, Coalescence, breakup and liquid circulation in bubble column reactors, Ph.D. thesis, Department of Chemical Engineering, The Norwegian Institute of Technology, Trondheim, Norway, 1993.

[40] M. Prince, H. Blanch, Bubble coalescence and breakup in air-sparged bubble columns, AIChE journal 36 (1990) 1485–1499.

[41] M. Laakkonen, V. Alopaeus, J. Aittamaa, Validation of bubble breakage, coalescence and mass transfer models for gas-liquid dispersion in agitated vessel, Chemical Engineering Science 61 (2006) 218–228.

[42] H. Luo, H. Svendsen, Theoretical model for drop and bubble breakup in turbulent dispersions, AIChE Journal 42 (1996) 1225–1233.

[43] M. Colombo, M. Fairweather, RANS simulation of bubble coalescence and break-up in bubbly two-phase flows, Chemical Engineering Science 146 (2016) 207 – 225.

[44] M. Colombo, R. Rzehak, M. Fairweather, Y. Liao, D. Lucas, Benchmarking of computational fluid dynamic models for bubbly flows, Nuclear Engineering and Design 375 (2021) 111075.

[45] D. Lucas, Y. Liao, R. Rzehak, E. Krepper, Towards best practice guidelines for Euler-Euler simulations of poly-disperse bubbly flows, in: NURETH-18.

[46] Agile delivery with GitLab, `https://about.gitlab.com/solutions/agile-delivery/`, 2020.

[47] M. D. Wilkinson, et al., The FAIR guiding principles for scientific data management and stewardship, Scientific data 3 (2016).

[48] Python workflow automation tool for time-series, `https://github.com/KIT-IAI/pyWATTS`, 2020.

[49] I. Evdokimov, S. Hänsch, F. Schlegel, Scalable Workflows for OpenFOAM Evaluation, in: IPS RAS 2020.

[50] D. Lucas, E. Krepper, H.-M. Prasser, Development of co-current airwater flow in a vertical pipe, International Journal of Multiphase Flow 31 (2005) 1304 – 1328.

[51] J. Köster, Snakemake report, `https://koesterlab.github.io/resources/report.html`, 2020.

[52] Github repository scikit-fuzzy, `https://github.com/scikit-fuzzy/scikit-fuzzy`, 2020.

[53] J. Warner, et al., scikit-fuzzy/scikit-fuzzy: Scikit-fuzzy 0.4.2, `https://zenodo.org/record/3541384`, 2019.

[54] S. S. Izquierdo, L. R. Izquierdo, Mamdani fuzzy systems for modelling and simulation: A critical assessment, Journal of Artificial Societies and Social Simulation 21 (2018) 2.

[55] H. Hessenkemper, T. Ziegenhein, R. Rzehak, D. Lucas, A. Tomiyama, Lift force coefficient of ellipsoidal single bubbles in water, International Journal of Multiphase Flow 138 (2021) 103587.

[56] S. Hänsch, M. Draw, I. Evdokimov, H. Khan, B. Krull, R. Lehnigk, Y. Liao, H. Lyu, R. Meller, F. Schlegel, M. Tekavčič, HZDR Multiphase Case Collection for OpenFOAM, `https://doi.org/10.14278/rodare.812`, 2021.

[57] S. Hosokawa, A. Tomiyama, Multi-fluid simulation of turbulent bubbly pipe flows, Chemical Engineering Science 64 (2009) 5308 – 5318.

[58] R. Rzehak, Y. Liao, R. Meller, F. Schlegel, R. Lehnigk, D. Lucas, Radial pressure forces in Euler-Euler simulations of turbulent bubbly pipe flows, Nuclear Engineering and Design 374 (2021) 111079.

[59] T. Liu, The role of bubble size on liquid phase turbulent structure in two-phase bubbly flow, in: 3rd International Conference on Multiphase Flow (ICMF98).

[60] S. Kriebitzsch, R. Rzehak, Baseline model for bubbly flows: Simulation of monodisperse flow in pipes of different diameters, Fluids 1 (2016) 29.

[61] M. Shawkat, C. Ching, M. Shoukri, Bubble and liquid turbulence characteristics of bubbly flow in a large diameter vertical pipe, International Journal of Multiphase Flow 34 (2008) 767 – 785.

[62] S. Hosokawa, A. Tomiyama, Bubble-induced pseudo turbulence in laminar pipe flows, International Journal of Heat and Fluid Flow 40 (2013) 97 – 105.

[63] Y. Liao, K. Upadhyay, F. Schlegel, Eulerian-Eulerian two-fluid model for laminar bubbly pipe flows: Validation of the baseline model, Computers & Fluids 202 (2020) 104496.

[64] M. Kim, J. Lee, H. Park, Study of bubble-induced turbulence in upward laminar bubbly pipe flows measured with a two-phase particle image velocimetry, Exp Fluids 57 (2016) 1432–1114.

[65] R. Lehnigk, W. Bainbridge, Y. Liao, D. Lucas, T. Niemi, J. Peltola, F. Schlegel, An open-source population balance modeling framework for the simulation of polydisperse multiphase flows based on the method of classes, AIChE Journal (submitted).