Master's Thesis

Evaluation of different drag models with the Multiphase Particle-In-Cell method in OpenFOAM

performed for the purpose of obtaining the academic degree of
Diplom Ingenieur under supervision of
Ass.Prof. Dipl.Ing. Dr.techn. Michael Harasek
and
Dipl.Ing. Markus Bösenhofer
and
Dipl.Ing. Eva-Maria Wartha
at the
Institute of Chemical, Environmental
and Bioscience Engineering, E 166
Faculty of Technical Chemistry
Technische Universität Wien
Master Programme Chemical
and Process Engineering
066 473

by

Alex Gludovacz
Matr.No. 01025075

05.04.2019

Affidavit

I confirm, that going to press of this thesis needs the confirmation of the examination committee.

I, Alex Gludovacz, declare in lieu of oath

1. that I wrote this thesis and performed the associated research myself, using only literature cited in this volume. If text passages from sources are used literally, they are marked as such.
2. that this work is original and has not been submitted elsewhere for any examination nor is it currently under consideration for a thesis elsewhere.

Vienna, 05.04.2019

Alex Gludovacz

Danksagung

Vielen Dank an Markus Bösenhofer, der mich überhaupt erst in die Welt der CFD Simulation eingeführt hat und mich damit auf die Idee gebracht hat, diese Arbeit zu schreiben. Seine gute Betreuung und besonders seine positive Einstellung haben mich immer wieder von neuem motiviert weiterzumachen. Besonderer Dank gilt auch Eva-Maria Wartha, die während Markus Auslandsaufenthalt meine Diplomarbeitsbetreuung anstandslos übernommen hat. Die beste Betreuung überhaupt.

Bedanken möchte ich mich auch bei Michael Harasek für die Möglichkeit, meine Diplomarbeit in seiner Arbeitsgruppe durchzuführen und für seinen fachlichen Rat.

Weiters bedanken möchte ich mich bei meinen Weggefährten auf dem Weg zum Abschluss, sei es zu Beginn im CFD-Kammerl oder auch gegen Ende im CFD-Büro.

Ganz besonderer Dank gilt auch allen Mitspielern und Gegnern beim Tischfußball, denn wenn ich in der Zeit hier an der Universität wirklich etwas gelernt habe, dann wuzeln. Besonders zu erwähnen sind hier die täglichen Duelle mit Mario.

Bedanken möchte ich mich auch bei meinen Kollegen aus dem 10er Jahrgang, die mich während des gesamten Studiums begleitet haben: Michi, Giuliàn, Dominik, Jeti, Gabriel, Ferdl, Voite und Hannes. Zusammen konnten wir schlussendlich alle Prüfungen meistern.

Dank gilt natürlich auch allen meinen Freunden ausserhalb des Studiums, die mit mir die lernfreie Zeit so unvergesslich gemacht haben. Sei es die Chaostruppe and Friends, die unguaden Puten, die Kabauter oder wie sie noch alle heißen. Ich bin echt froh sie zu haben.

Natürlich möchte ich mich auch bei meiner Familie für die stete Unterstützung bedanken. Ohne so einen starken Rückhalt und Zuspruch in schlechteren Phasen hätte ich das Studium wohl nicht geschafft.

Schlussendlich möchte ich mich noch beim besten Menschen bedanken, den ich kenne: Danke Meli.

Abstract

Current blast furnace technology exists since the nineteenth century and has always been one of the biggest emitters of carbon dioxide. A blast furnace is used for smelting iron ore to produce pig iron. A hot air blast is injected at the lower part of the blast furnace. The raceway is a turbulent zone, that occurs when hot air is blown into the coke filling and is a significant part of the blast furnace. Although it has been widely studied, not all processes happening inside the raceway are known yet. CFD simulation can be a useful tool for improving the efficiency of blast furnaces for a more climate-friendly steel production.

The blast furnace process is kind of a multiphase flow, with particles dispersed in fluid phases. There are two major modeling concepts for handling fluid-particle flows: Eulerian-Eulerian (EE) and the coupled CFD - discrete elements method (CFD-DEM). As a hybrid method, Multiphase Particle-In-Cell (MP-PIC) shall combine the advantages of both. Single particles are combined to so-called parcels. These parcels are supposed to behave like the equivalent number of real particles. Therefore, the particle number inside the system and the computational effort are reduced, while the accuracy stays acceptable.

An important part of any gas-particle flow is the drag model. It describes the momentum exchange between particles and the surrounding fluid. Different approaches describe the exchange of forces and moments.

In this work, a comparison of drag models from literature was conducted using typical raceway conditions. Additionally, the drag model performance was evaluated by comparing the predicted pressure drop to experimental values. In the next step, Eulerian-Lagrangian (EL) hybrid models in combination with chosen drag models were evaluated for a fixed bed and a fluidized bed. Finally, the capabilities to predict raceway formation were investigated using experimental literature.

The simulations showed, that the Richardson-Zaki model failed to predict reasonable results for the first test case and was excluded from the subsequent evaluations. The remaining models predict reasonable pressure drops. EL hybrid models showed good agreement for the fixed bed test case, while the results depend on the drag models for the fluidized bed case. The EL models failed to predict the raceway formation for one of the two investigated cases. In the second simplified case, the EL models gave acceptable results.

Findings of these simulations can be used for developing EL hybrid models for the raceway zone.

Kurzfassung

Hochöfen bestehen in ähnlicher Form wie heute seit dem 19. Jahrhundert und gehören seit jeher zu den größten Emittenten von Kohlendioxid. In einem Hochofen wird aus Eisenerz in einem kontinuierlichen Reduktionsprozess Roheisen erzeugt. Die Raceway ist die Wirbelzone, die beim Einblasen von Heißluft in die Kohleschüttung entsteht und ist ein integraler Bestandteil des Hochofenvorgangs. Trotz intensiver Forschung sind noch immer nicht alle Vorgänge darin verstanden. CFD Simulation kann ein nützliches Hilfsmittel zur Verbesserung der Effizienz von Hochöfen auf dem Weg zu einer klimafreundlicheren Stahlproduktion sein.

Der Hochofenprozess lässt sich als Mehrphasenströmung beschreiben, da Partikel in Fluidphasen feinverteilt sind. Zur Simulation von Partikelströmungen gibt es zwei grundlegende Ansätze, die Euler-Euler Modelle(EE) und die CFD-Diskrete Elemente Methode (CFD-DEM). Multiphase Particle-In-Cell (MP-PIC) soll als Hybridmodell die Vorzüge von beiden verbinden. Dabei werden Partikel in sogenannte Pakete zusammengefasst. Diese Pakete sollen sich wie die äquivalente Anzahl von realen Partikeln verhalten. Dadurch wird die Anzahl der Partikel im System reduziert und der Rechenaufwand bei gleichbleibend akzeptabler Genauigkeit verringert.

Ein wichtiger Bestandteil von Partikelströmungssimulationen ist das Widerstandsmodell, welches den Strömungswiderstand eines Partikels in einer Fluidströmung beschreibt. Verschiedene Ansätze beschreiben den Impulsaustausch.

In dieser Arbeit wurde zuerst ein Vergleich von in der Literatur bereits existierenden Widerstandsmodellen anhand für eine Raceway typischen Bedingungen durchgeführt. Zusätzlich wurden die Widerstandsmodelle durch einen Vergleich des berechneten Druckverlusts mit experimentiellen Daten getestet. Im nächsten Schritt wurden die Euler-Lagrange (EL) Hybridmodelle in Kombination mit den gewählten Widerstandsmodellen für die Verwendung in einem Festbett und einer Wirbelschicht evaluiert. Als Letztes wurde die Fähigkeit der Modelle eine Racewaybildung vorauszuberechnen untersucht.

Die Berechnungen zeigten, dass das Richardson-Zaki Modell keine vernünftigen Ergebnisse für den ersten Testfall geliefert hat. Deshalb wurde es nicht in den weiteren Tests berücksichtigt. Die restlichen Modelle zeigten vernünftige Druckverluste. Die EL Hybridmodelle zeigten eine gute Übereinstimmung für den Festbett Testfall. Die Ergebnisse des Wirbelschicht Testfalls hängen jedoch vom Widerstandsmodell ab. Die EL Modelle haben beim Berechnen der Racewaybildung im ersten der beiden Testfälle versagt. Im zweiten Fall lieferten sie akzeptable Ergebnisse.

Die Ergebnisse dieser arbeit können für die Entwicklung von EL Hybridmodellen für die Raceway Zone verwendet werden.

Contents

List of Abbreviations

CFD	computational fluid dynamics
CFD-DEM	computaional fluid dynamics - discrete elements method
DEM	discrete elements method
DNS	direct numerical simulation
DPM	discrete phase method
EE	Eulerian-Eulerian
EL	Eulerian-Lagrangian
KTGF	kinetic theory of granular flows
LBM	Lattice Boltzmann method
LHS	left hand side
MP-PIC	multiphase particle-in-cell method
HKL	Hill-Koch-Ladd
OpenFOAM	Open source Field Operation and Manipulation
PIC	particle-in-cell
PIMPLE	PISO-SIMPLE
PISO	Pressure Implicit with Splitting of Operator
PUReIBM	Particle-resolved Uncontaminated - fluid Reconcilable Immersed Boundary Method
RHS	right hand side
RUC	Representative Unit Cell
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations

List of Symbols

Symbol	**Unit**	**Description**
$\vec{A}$	ms^{-2}	acceleration
A	m^2	area
A, B	$-$	coefficient depending on the solid volume fraction
C_D	$-$	drag coefficient
d	m	diameter
e	$-$	coefficient of restitution
F	$-$	dimensionless drag factor
$\vec{F}$, $\vec{f}$	N	force
$\vec{F}$	N	virtual mass force
f	$-$	function
F_d	N	drag force
$g, \vec{g}$	$m\ s^{-2}$	gravitational acceleration
g_o'	$-$	radial distribution function
K	$kg\ m^{-3}\ s^{-1}$	momentum transfer coefficient
$K_{pq}\vec{v}_{dr}$	N	turbulent dispersion force
L	m	charactersitic length scale
Ma	$-$	Mach number
m	kg	mass
n	$-$	Richardson-Zaki-parameter
p	$N\ m^{-2}$	pressure
P_s	$N\ m^{-2}$	solid phase stress
r	m	radius
Re	$-$	Reynolds number
r_{32}	m	particle Sauter mean radius
$r_p(t)$	ms^{-1}	translational velocity
S	$-$	source
St	$-$	Stokes number
$u, \vec{u}, v$	$m\ s^{-1}$	flow speed, velocity
V	m^3	volume
α	$-$	volume fraction
ϵ	$-$	threshold parameter, e.g. 10^{-7}
η	$-$	dependence on the coefficient of restitution
μ	$Pa \cdot s$	dynamic viscosity
ν	$m^2\ s^{-1}$	kinematic viscosity
ξ	$-$	probability distribution function
ρ	$kg\ m^{-3}$	mass density
σ	$N\ m^{-2}$	stress tensor
τ	s	time

Symbol	Unit	Description
τ	–	particle stress
ϕ	–	shape factor (1 corresponds to perfect sphere)
ψ	–	auxiliary coefficient for the HKL drag model
ω	–	auxiliary coefficient for the HKL drag model
∇	–	nabla operator

Super-/Subscript	Description
$1, 2, 3, ...$	numeration
B	Basset
$Beet$	Beetstra
$cell$	cell of the computational mesh
$char$	characteristic
cp	close pack
D	drag
dim	dimensionless
dis	distribution
dr	drift
ext	external pressure
f	fluid
hp	hydrostatic pressure
HKL	Hill-Koch-Ladd
i, j	phase i, j
$isol$	isolated
lim	limiting
$lift$	lift
$mass$	mass
$mean$	mean
mom	momentum
n	number
p	particle
$parcel$	parcel
r	radius
res	response
rel	relative
RZ	Richardson-Zaki
s	solid (particle)
$spher$	spherical
Ten	Tenneti
td	turbulent dispersion
$trans$	transversal
$virt$	virtual
β	constant
$-$	averaged

1 Introduction

1.1 Motivation and Problem Statement

The blast furnace technology was developed in a span of more than 2000 years [Golas, 1999]. Since the nineteenth century, blast furnaces are operated with coke and air as hot blast. Since the early days, the iron industry has been one of the major emitters of carbon dioxide. A better understanding of the many processes happening inside a blast furnace is helpful for building more efficient and environmental friendly plants. Simulations can help there. They can be an effective and relatively cheap tool for understanding the different involved processes in more detail and reveal non-observable processes.

In a blast furnace, iron ore, coke and flux are inserted through the top of the furnace, while hot blast (oxygen enriched air) is blown into the bottom of the blast furnace through so-called tuyeres. A raceway is formed in the vicinity of the tuyeres, where extreme conditions prevail: high velocities, high temperatures, elevated pressure, featuring chemical reactions including fluid phases and particles.

There are different ways for the simulation of particle flows in the blast furnace [Jakobsen, 2008]. One way is the Eulerian-Eulerian (EE) approach, were particulate phases are approximated as continuous phase using the kinetic theory of granular flows (KTGF) [Gidaspow, 1994]. The EE framework is less computationally expensive, but shows a lack of accuracy [Adamczyk et al., 2014]. Another approach is the computational fluid dynamics-discrete elements method (CFD-DEM), where the motion of distinct particles is tracked in time and space using the Lagrangian approach. [Norouzi et al., 2016]. An advantage of these models is their ability to display particle-particle interactions in detail. The major disadvantage is the significantly increased computational effort compared to the EE framework. DEM is a powerful and exact simulation technique, but the explicit integration of its equations for large numbers of interacting particles demands a huge computational effort [Norouzi et al., 2016].
Hybrid models combine the advantages of both methods. They use the Eulerian-Lagrangian (EL) approach, where the Eulerian approach is used for the continuous phase and the Lagrangian approach is used for the solid/discrete phase. These models shall combine the benefits of EE and DEM. Two existing hybrid models

to simulate particle flows are the discrete phase method (DPM) [Adamczyk et al., 2014] and the multiphase particle-in-cell method (MP-PIC) [Andrews and O'Rourke, 1996], further developed by [Snider, 2001], [O'Rourke and Snider, 2010], [O'Rourke and Snider, 2012]. The computational effort is reduced in these models through combining particles to so-called parcels. These parcels should behave like the equivalent number of real particles. DPMFoam and MPPICFoam are the solvers implemented in the open-source CFD code OpenFOAM [Weller et al., 1998] representing the particular concepts.

A drag model is an important part of multiphase simulations. It describes the momentum exchange between particles and the surrounding fluid. Different drag models have been developed. Some are derived especially for certain conditions, but the presented models claim to be valid for a wide range of Reynolds numbers and solid volume fractions. They are compared to each other and used in the solver evaluations.

1.2 Objectives of this Work

The objective of this work is the evaluation of the ability of the DPM and MPPIC approach to predict blast furnace-like gas-solid flows. Both approaches should reduce the numerical effort compared to CFD-DEM. The DPM approach is used when cell sizes are at the same scale as the particle sizes, since particle-particle interactions are resolved. The work of [Haddadi et al., 2017] provides such a test case for the DPM approach.

In contrast, the MP-PIC method solves the particle interactions on the Eulerian grid. This approach reduces the numerical effort, but requires significantly larger cells than particles. The work of [Jayarathna et al., 2017] provides a suitable test case for the MP-PIC evaluations in slowly moving beds. MP-PICFoam's actual capabilities to predict the raceway formation are given by [Mojamdar et al., 2018]. Additional literature for a second raceway validation is given by [Feng et al., 2003].

2 Theory

2.1 Multiphase Flow

The term multiphase flow is used for any fluid flow that consists of more than one phase or component [Brennen, 2005]. This can either be a simultaneous flow of

- materials with different states or phases (i.e. gas, liquid or solid), or
- materials with different chemical properties but in the same state or phase (i.e. liquid-liquid systems such as oil droplets in water)

Examples for multiphase flow in nature are rain, snow, fog, avalanches, mud slides, etc. [Brennen, 2005].
Every phase has separately defined volume fraction and velocity fields. Figure 2.1 shows typical multiphase flow regimes identified in general gas-liquid and gas-solid flows in vertical and horizontal tubes. They can vary from simple slug and bubbly flows (A, B) to dispersed systems like particulate flow (F).

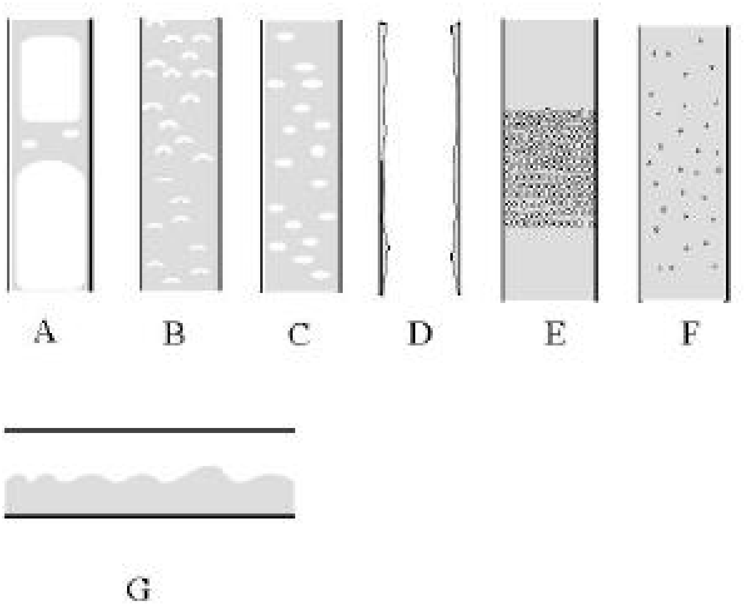


Figure 2.1: Multiphase flow regimes. A: Slug flow. B: Bubbly flow. C: Droplet flow. D: Annular flow. E: Packed and porous fixed bed. F: Particulate flow. G: Stratified-free surface flow. [Jakobsen, 2008]

The following list shows typical occurrences of the different flow regimes in a chemical reactor [Jakobsen, 2008]:

- Single phase fluid flows (single phase reactors, i.e., laminar and turbulent flows)
- Flow through porous beds of solids (fixed bed reactors)
- Granular flows (fluidized bed reactors)
- Bubbly flows (fluidized bed reactors)
- Slurries (mixture of a pulverized solid with a liquid)
- Complex multiphase flows, where many phases interact simultaneously.

2.2 Modeling Concepts for Multiphase Flow

When dealing with simulations for multiphase systems, one has to differ between systems with separated flows (regime G in Figure 2.1) and systems with dispersed flows (regimes B, C and F in Figure 2.1). Different computational strategies for multiphase flows are available. They are described in detail in the following sections.

2.2.1 The Eulerian-Eulerian Multifluid Models

In the EE framework, particulate phases are approximated as continuous phase by the KTGF [Jakobsen, 2008].
These models use a set of transport equations including the conservation equation of mass and momentum.
The governing equations are expressed as [Adamczyk et al., 2014]:

$$\frac{\partial}{\partial t}(\alpha_i \rho_i) + \bigtriangledown \cdot (\alpha_i \rho_i \vec{u}_i) = 0 \tag{2.1}$$

$$\frac{\partial}{\partial t}(\alpha_f \rho_f \vec{u}_f) + \bigtriangledown \cdot (\alpha_f \rho_f \vec{u}_f \vec{u}_f) = -\alpha_f \bigtriangledown p + \bigtriangledown \cdot \tau_f + \alpha_f \rho_f \vec{g} + F + \sum_{q=1}^{N} [K_{if}(\vec{u}_f - \vec{u}_i)] \tag{2.2}$$

$$\frac{\partial}{\partial s}(\alpha_s \rho_s \vec{u}_s) + \bigtriangledown \cdot (\alpha_s \rho_s \vec{u}_s \vec{u}_s) = -\alpha_s \bigtriangledown p + \bigtriangledown \cdot \sigma_s + \alpha_s \rho_s \vec{g} + F + \sum_{q=1}^{N} [K_{is}(\vec{u}_i - \vec{u}_s)] \tag{2.3}$$

The derivation of the governing equations can be found in [Gidaspow, 1994].
Equation 2.1 describes the continuity equation for a phase i, whereas Equation 2.2 and Equation 2.3 define the momentum conservation for fluid (f) and solid (s) phases.

According to [Jakobsen, 2008], the multi-fluid model can be used to describe any multiphase flow regime, when the closure relation for the interfacial coupling terms are provided. Additionally, the EE approach is less computationally expensive than the Eulerian-Langrangian (EL) approach (see subsection 2.2.3). However, there is no general agreement on the correct mathematical form of the net interfacial forces acting on the dispersed phase particles, nor on the net hydrodynamic and fluid-particle turbulence interactions. However, in recent years promising progress has been made though [Jakobsen, 2008].

2.2.2 CFD-DEM

The Computational Fluid Dynamics - Discrete Element Method (CFD-DEM) is used to describe and compute systems of fluids and large numbers of particles. In the multiphase flow procedure, fluid and solid phases can be modeled by the EE approach, and in computaional fluid dynamics - discrete elements method (CFD-DEM), the motion of distinct particles is tracked in time and space using the Lagrangian approach [Norouzi et al., 2016].

The basic principle of a DEM simulation is that matter consists of individual, separate (discrete) elements. In granular flow, these elements are particles. An initial velocity is given to the particles at the beginning of a DEM simulation. Particles can interact with each other and their contact lasts for a certain period. This allows particles to interact with more than one other particle at once. All motion force equations in DEM are solved explicitly, which increases the flexibility of DEM simulations, but requires a small time step for integration. This small time step in combination with a large number of involved particles demands huge computational effort. [Norouzi et al., 2016]

CFD-DEM is settled between the meso-scale of the continuum and the micro-scale of the particles, visible in Figure 2.2. In this theory, the flow field is separated into cells which are larger than the particles but smaller than the flow field. The volume fraction of each phase and the momentum exchange through the drag force are the main influences on the motion of the particles. The fluid phase in meso-scale is described by the local averaged Navier-Stokes equations via the traditional computational fluid dynamics (CFD) approach, while the motion of solid particles in micro-scale is obtained by the DEM, which applies Newton's and Euler's second laws to every particle. A CFD-DEM simulation is still computationally expensive and has therefore limitations to the total cell and particle number and the time step of integration [Norouzi et al., 2016].

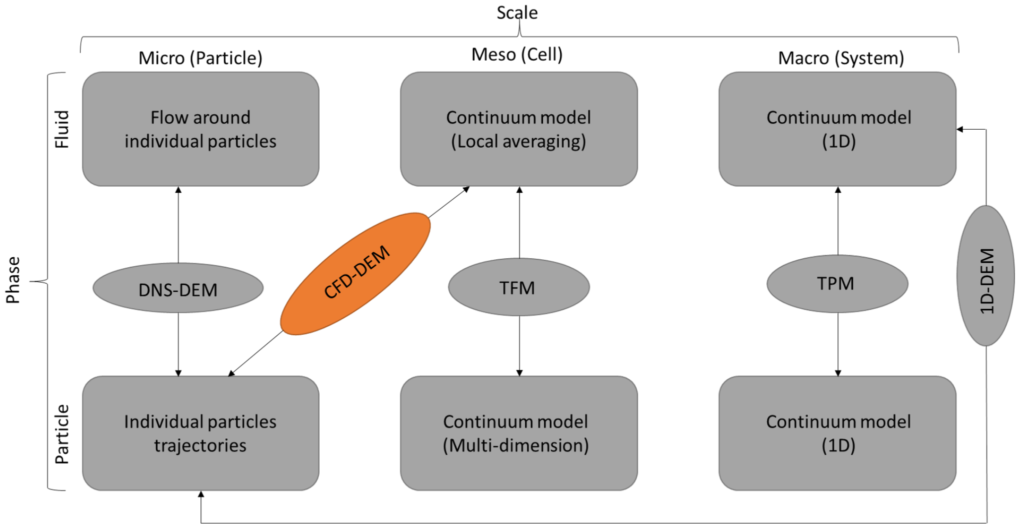


Figure 2.2: Modeling scales in fluid-particulate systems according to [Norouzi et al., 2016]. TFM: two-fluid model, TPM: two-phase model

2.2.3 The Eulerian-Lagrangian Models

Eulerian-Lagrangian (EL) models use the Eulerian approach for the fluid phase and the Lagrangian approach for the solid/discrete phase. They are based on Langrangian mechanics, which are a reformulation of classical mechanics. Langrangian mechanics is also true in a non-inertial reference frame and invariant for coordinate transformation [Honerkamp et al., 2012].
In the Lagrangian framework, particles are treated as rigid, infinitesimal small spheres, similar to point centers of mass in space. [Jakobsen, 2008]. Equation 2.4 describes the Langrangian form of Newton's second law, which is used to describe the translational motion of the particle:

$$\frac{d}{dt}(m_p\vec{v}_p) = \vec{f}_{hp} + \vec{f}_{ext} + \vec{f}_E + \vec{g} + \vec{f}_D + \vec{f}_{virt} + \vec{f}_{translift} + \vec{f}_B \tag{2.4}$$

With:
$\vec{f}_{hp}$ ... force due to hydrostatic pressure
$\vec{f}_{ext}$ ... force due to any external pressure gradients
$\vec{f}_E$ ... body force created by external fields apart from gravity
$\vec{f}_D$ ... steady drag force
$\vec{f}_{virt}$ ... virtual mass force
$\vec{f}_{translift}$ ... transversal lift force
$\vec{f}_B$ ... Basset history force

and $m_p = \rho_p V_p$ describing the particle mass. Particle trajectories are calculated from the definition of the translational velocity $r_p(t)$ of the particle mass center:

$$\frac{d\vec{r}_p(t)}{dt} = \vec{v}_p(t, \vec{r}_p(t)) \tag{2.5}$$

EL concepts are classified according to the employed phase coupling [Jakobsen, 2008]:

- In a *one-way coupled* system, the particle volume fraction is in a range where the effects of the particulate phase on the continuous phase are negligible. However, the local velocity of the continuos phase has a direct impact on the particle movement. For these systems, the Eulerian velocity field is computed independently of the particle tracking with a standard single-phase simulation. Thus, the trajectories of the single particles are also computed independently from each other.

- *Two-way coupled* systems, in contrast, consider the phase coupling for fluid and particulate phases. Two way coupling is required for low to moderate particle phase fractions.

- For dense systems, *four-way coupling* has to be used, because additional to inter-phase momentum exchange, particle-particle interactions are important. Four-way coupling is required for high solid phase fractions.

Obviously, the number of particles and the volume fraction dictate the system of choice. In practice, for dilute systems ($\alpha_p < 10^{-6}$) one way coupling is sufficient. Two way coupling is required for $\alpha_p < 10^{-3}$ and for denser systems, four way coupling is necessary [Jakobsen, 2008].

Figure 2.3 shows the classification of phase-coupling mechanisms and Figure 2.4 gives a visual representation of the interactions.

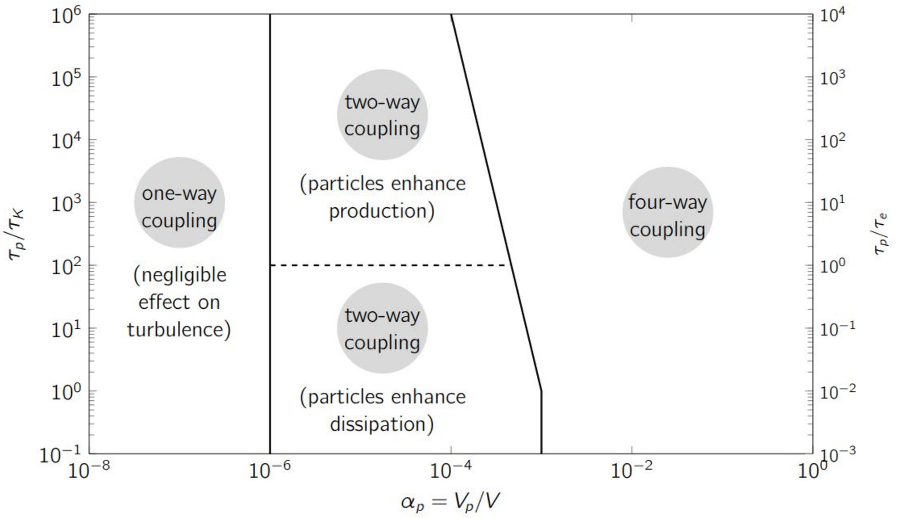


Figure 2.3: Classification of phase-coupling mechanisms according to [Elghobashi, 1994]

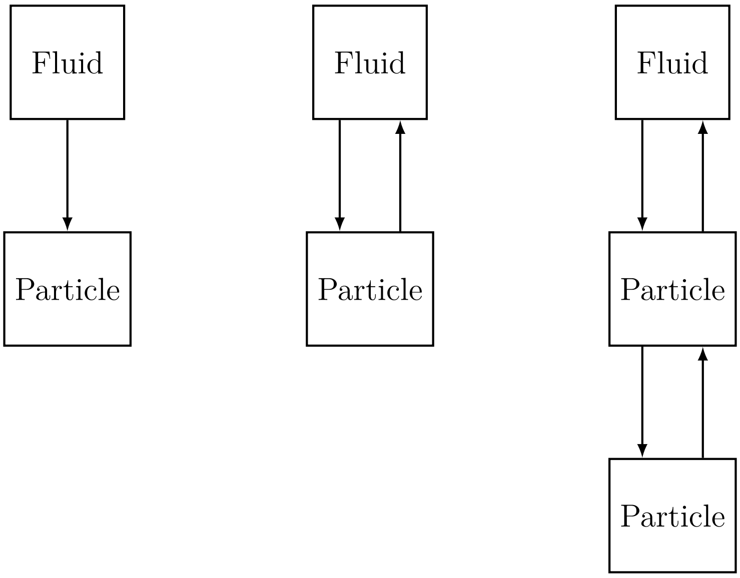


Figure 2.4: Visual representation of possible phase interactions. left: one way coupling, center: two way coupling, right: four way coupling

The main advantage of the EL framework is its ability to describe the particle-particle interactions in detail. While the major disadvantage is the increased computational effort compared to the EE framework. Simple EL cases can be solved with a powerful personal computer within minutes, whereas complex 3D-cases have to be solved on supercomputers [Jakobsen, 2008].

2.2.3.1 Discrete Phase Method

The DPM is an Eulerian-Lagrange approach using a four-way coupling for describing particle flow, as described in the previous sections. The numerical solution of the particle phase is obtained by concentrating a large number of particles to a so called parcel. This parcel is supposed to behave like the equivalent number of real particles with the same density, volume, velocity and position. The number of single particles in a computational parcel can be determined by following expression [Adamczyk et al., 2014]:

$$n_P = \frac{\dot{m}_{parcel}\Delta t}{m_P}, \tag{2.6}$$

where:
Δt … time step in a transient calculation
$\dot{m}_{parcel}$ … mass flow rate of a single parcel
m_P … mass of an individual particle.

Following expressions describe the equations for mass and momentum solved in the

DPM approach:

$$\frac{\partial}{\partial t}(\alpha_f \rho_f) + \nabla \cdot (\alpha_f \rho_f \vec{u}_f) = S_{mass} \tag{2.7}$$

$$\frac{\partial}{\partial t}(\alpha_f \rho_f \vec{u}_f) + \nabla \cdot (\alpha_f \rho_f \vec{u}_f \vec{u}_f) = -\alpha_f \nabla p + \nabla \cdot \tau_f + \alpha_f \rho_f \vec{g} + K_{sf}(\vec{u}_s - \vec{u}_f) + S_{mom}, \tag{2.8}$$

where S_{mass} and S_{mom} are terms for the exchange of mass and momentum between the particle and the fluid phase.

The particle equation of motion is defined as

$$\frac{d\vec{u}_p}{dt} = F_D(\vec{u}_f - \vec{u}_p) + \frac{\vec{g}(\rho_p - \rho_f)}{\rho_p} - \frac{\nabla p}{\rho_p} - \frac{\nabla \cdot \sigma_s}{\rho_p}. \tag{2.9}$$

In this equation, the granular stress tensor σ_s portrays the interactions calculated based on the KTGF [Gidaspow, 1994] on the Eulerian grid, and the term $F_D(\vec{u}_f - \vec{u}_p)$ represents the particle acceleration due to drag. $\nabla p/\rho_p$ describes the particle acceleration due to pressure difference. A new position of a particle is calculated as

$$\frac{d\vec{x}_p}{dt} = \vec{u}_p. \tag{2.10}$$

After the calculation of the particle position, the solid volume fraction in a computational cell is calculated as follows:

$$\alpha_s = \frac{n_p n_{parcel} V_p}{V_{cell}}, \tag{2.11}$$

where V_{cell} is the volume of a computational cell. The obtained solid volume fraction is transferred to the Eulerian grid where the fluid volume fraction is calculated using the correlation $\alpha_f = 1 - \alpha_s$ [Adamczyk et al., 2014]. Section 2.2.3.2 gives detailed information about the solid stress tensor σ_s. More information about the derivation of these governing equations can be found in literature [Gidaspow, 1994].

2.2.3.2 Multiphase-Particle In Cell

The MP-PIC method is a numerical method for calculating particle-particle and particle-fluid interaction in CFD [Andrews and O'Rourke, 1996]. Its predecessor was the particle-in-cell (PIC) method. The numerical solution of the particle phase is obtained by concentrating a large number of particles to so-called parcels. These parcels are supposed to behave like the equivalent number of real particles with the same density, volume, velocity and position. In the MP-PIC, the particles are treated as computational particles and as continuum at the same time. The MP-PIC core method is the mapping of particle properties between the Lagrangian and Eulerian phase via interpolation functions. In the solution algorithm, particle properties are

first solved on the Eulerian grid and then mapped back to the particle parcels prior to solving the Lagrangian phase. [Andrews and O'Rourke, 1996]
For the MP-PIC and an incompressible gas phase, following continuity equation emerges:

$$\frac{\partial \alpha_f}{\partial t} + \nabla \cdot (\alpha_f \overrightarrow{u_f}) = 0 \tag{2.12}$$

A variation of the Navier-Stokes equations describes the momentum transport:

$$\frac{\partial \alpha_f \overrightarrow{u_f}}{\partial t} + \nabla \cdot (\alpha_f \overrightarrow{u_f} \overrightarrow{u_f}) = -\frac{\nabla p}{\rho_f} - \frac{\vec{F}}{\rho_f} + \alpha_f \vec{g} \tag{2.13}$$

The particle phase is described by a so called probability distribution function (ξ), which indicates the likelihood of finding a particle with a velocity $\overrightarrow{u_f}$, density ρ_p, volume V_p at location x and time t. Following equation describes its changes in time:

$$\frac{\partial \xi}{\partial t} + \nabla \cdot (\alpha_g \overrightarrow{u_p}) + \nabla \overrightarrow{u_p} \cdot (\alpha_g \vec{A}) = 0 \tag{2.14}$$

where $\vec{A} = \frac{d\vec{u}}{dt}$ is the particle acceleration and is given by

$$\vec{A} = D(\overrightarrow{u_f} - \overrightarrow{u_p}) - \frac{1}{\rho_p} \nabla_x p + \vec{g} - \frac{1}{\alpha_s \rho_s} \nabla_x \tau \tag{2.15}$$

where D is a drag coefficient (see section 2.3) and τ is an isotropic solid stress term that will be discussed later. The particle volume fraction is described by following equation, which includes the distribution function (f_{dis}):

$$\alpha_s = \iint f_{dis} \frac{m_p}{\rho_s} dm_p \, d\vec{v}. \tag{2.16}$$

The interface momentum transfer function is provided by following expression:

$$\vec{F}_{transfer} = \iint f_{dis} m_p \left[D(\overrightarrow{u_f} - \overrightarrow{u_p}) - \frac{1}{\rho_s} \nabla \vec{p} \right] dm_p \, d\vec{v}. \tag{2.17}$$

The computational parcels follow characteristic paths representing the positions, velocities and sizes of physical particles. Particle equations are not directly solved, but as mentioned before, are interpolated to the Eulerian grid. The solved equations are of the following form [Travis and Harlow, 1976]:

$$\frac{\partial (\alpha_s \rho_s)}{\partial t} + \nabla_x \cdot (\alpha_s \rho_s \vec{u}_s) = 0 \tag{2.18}$$

$$\begin{aligned}
&\frac{\partial(\alpha_s\rho_s\vec{u}_s)}{\partial t} + \nabla_x\cdot(\alpha_s\rho_s\vec{u}_s\vec{u}_s) + \nabla_x\tau + \alpha_s\nabla_x p = \\
&\alpha_s\rho_s\vec{g} + \iint f_{dis}m_p D(\vec{u}_g - \vec{u}_s)dm_p\; d\vec{v} - \\
&\nabla_x\left[\iint f_{dis}m_p(\vec{u}_s\vec{u}_{p,mean})(\vec{u}_s\vec{u}_{p,mean})dm_p\; d\vec{v}\right]
\end{aligned} \tag{2.19}$$

where $\vec{u}_{p,mean}$ describes the mean particle velocity:

$$\vec{u}_{p,mean} = \frac{1}{\alpha_s\rho_s}\iint f_{dis}m_p\;\vec{v}\;dm_p\;d\vec{v}. \tag{2.20}$$

After the Eulerian particle phase equations are solved, the gradients of velocities, gas pressure and solid stress are interpolated back to the parcel positions. This and the fact that there are typically less grid points than Lagrangian parcels save a lot of computational cost. [Andrews and O'Rourke, 1996]

The interpolation of the particle properties is done with interpolation operators. A rectilinear grid is assumed and the particle properties are mapped to the cell centers, while the vector properties are mapped to the cell faces. [Snider, 2001]

The effect of particle packing is modeled through particle stress models. This provides a pressure type force and prevents packing of particles beyond the close pack limit. [Snider, 2001] has suggested calculating the particle stress τ, as

$$\tau = \frac{P_s\tau_p^{\beta}}{max\left[\alpha_{cp}-\alpha_s,\;\epsilon(1-\alpha_s)\right]}, \tag{2.21}$$

where:
α_{cp} ... close pack porosity
β ... constant, $2 \leq \beta \leq 5$
P_s ... solid phase stress
ϵ ... very small number used to avoid singularity, typically 10^{-7}.

The MP-PIC method was adapted by a model for the numerical calculation of collisional transfer of mass, momentum, and energy among particles in fluidized beds [O'Rourke et al., 2009]. Additional improvements regarding the particle-collision-behavior have been done by introducing the so-called damping time τ_D [O'Rourke and Snider, 2010]:

$$\tau_D = \frac{\sqrt{3\pi}}{32}\frac{d}{\alpha_s\sigma}\frac{1}{g_o'\eta\;(1-\eta)}, \tag{2.22}$$

where

$$g_o'(\alpha_s) = \frac{1}{1-\left(\frac{\alpha_s}{\alpha_{cp}}\right)^{\frac{1}{3}}} \tag{2.23}$$

describes the radial distribution function [Lun et al., 1984], and

$$\eta = \frac{1+e}{2} \tag{2.24}$$

is the dependence on the restitution coefficient (e).
Equation 2.22 is obtained from a correlation for the damping term by [Igci et al., 2008].

[O'Rourke and Snider, 2012] improved this formula by including collisional return-to-isotropy:

$$\frac{1}{\tau_g} = \frac{8\sqrt{2}}{5\pi}\frac{s}{r_{32}^3}\frac{\iint f(r+r_{32})^4(u_i-\bar{u_i})^2 dm_p\ dv_i}{\iint f(r+r_{32})^2(u_i-\bar{u_i}) dm_p\ dv_i} g_0(\alpha_s)\eta(2-\eta) \tag{2.25}$$

Where $\bar{u_i}$ describes the mass-averaged particle velocity. r and r_{32} describe the particle radius and the particle Sauter mean radius:

$$r = \left(\frac{m_p}{\frac{4}{3}\pi\rho_p}\right)^{\frac{1}{3}} \tag{2.26}$$

$$r_{32} = \frac{\int f r^3\ dm_p\ dv_i}{\int f r^2\ dm_p\ dv_i} = 6\frac{V_p}{A_p}. \tag{2.27}$$

These improvements are already implemented in the OpenFOAM code in version 6. The Usage of these models is described in subsection 3.1.2.

2.3 Multi-Phase Momentum Exchange

The drag coefficient is a dimensionless quantity that quantifies the intensity of the momentum exchange between an object in a fluid environment. It is used in the momentum equation in which a lower drag coefficient indicates that the object will have less aerodynamic or hydrodynamic drag. The drag coefficient is always associated with a particular surface area [McCormick, 1979].
The drag coefficient is commonly denoted as C_D and is defined as:

$$C_D = \frac{2 \cdot F_D}{\rho_f \cdot |\vec{u}_s - \vec{u}_f|^2 \cdot A}, \tag{2.28}$$

where:
F_D . . . drag force
ρ_f . . . fluid density
$|\vec{u}_s - \vec{u}_f|$ . . . relative velocity magnitude
A . . . surface area.

C$_D$ is a function of the Reynolds number Re for incompressible fluids:

$$C_D = f(Re), \tag{2.29}$$

with

$$Re = \frac{u \cdot \rho \cdot L}{\mu}, \tag{2.30}$$

where:
L . . . characteristic length scale of the object
μ . . . dynamic viscosity.

The drag force on any object is a function of density, flow speed, fluid viscosity, and characteristic length scale of the object:

$$F_D = f(\rho, u, \mu, L). \tag{2.31}$$

In compressible flows, the speed of sound is relevant and C$_D$ is also a function of the Mach number (Ma):

$$C_D = f(Re, Ma). \tag{2.32}$$

2.4 Drag Model Correlations

The following section gives an overview of selected drag models. For comparison, the so called momentum transfer coefficient K_{sf} is introduced. Its definition varies for the different drag models in the following sections. The unit of the momentum transfer coefficient is:

$$[K_{sf}] = \frac{kg}{m^3 \cdot s}. \tag{2.33}$$

2.4.1 Syamlal O'Brien

The Syamlal O'Brien drag model [Syamlal and O'Brien, 1987] was derived for a single spherical particle in a fluid and is modified with the relative velocity correlation v_{rel}, which describes the settling velocity of a particle divided by the settling velocity of a perfect sphere with identical mass [Lundberg J., 2008].
The main idea in this model is that the Archimedes numbers of a single particle and a multi-particle system are the same. The Archimedes number is a dimensionless number which describes the ratio of buoyancy forces and frictional forces [Dittmann, 1995]. The Archimedes number is defined by:

$$Ar = \frac{gL^3\rho_f(\rho_s - \rho_f)}{\mu^2}, \tag{2.34}$$

where:
g . . . gravitational acceleration
μ . . . dynamic viscosity
L . . . characteristic length scale of the object
ρ_s . . . density of the solid phase.

Following correlation shows the overall momentum exchange rate:

$$K_{sf} = \frac{3\alpha_f\alpha_s\rho_f}{4d_p v_{rel}^2} \cdot C_D|\vec{u}_s - \vec{u}_f|, \tag{2.35}$$

where:
α_f . . . volume fraction of the fluid phase
α_s . . . volume fraction of the solid phase
v_{rel} . . . relative velocity correlation
d_p . . . particle diameter.

Equation 2.36 shows the employed correlation for C_D, which was first proposed by [Dalla Valle, 1943]:

$$C_D = \left[0.63 + \frac{4.8}{\sqrt{\frac{Re_p}{v_{rel}}}}\right]^2 . \tag{2.36}$$

v_{rel} was first described by [Garside and Al-Dibouni, 1977]. [Richardson and Zaki, 1997] adapted the concept to yield the following expression:

$$v_{rel} = \frac{1}{2}\left[A - 0.06Re_p\right] + \frac{1}{2}\left[\sqrt{(0.06Re_p)^2 + 0.12Re_p(2B - A) + A^2}\right], \tag{2.37}$$

with the coefficients A and B being:

$$A = \alpha_f^{4.14} \tag{2.38}$$

$$B = \begin{cases} 0.8{\alpha_f}^{1.28} & \alpha_f \leq 0.85 \\ {\alpha_f}^{2.65} & \alpha_f > 0.85, \end{cases} \tag{2.39}$$

and Re_p describing the particle Reynolds number:

$$Re_p = \frac{\rho_f d_p |\vec{u}_s - \vec{u}_f|}{\mu_f}. \tag{2.40}$$

2.4.2 Richardson Zaki

The Richardson and Zaki model [Richardson and Zaki, 1997] is identical to the model proposed by Syamlal O'Brien, except for the definition of v_{rel}. An iterative algorithm defines v_{rel} in case of the Richardson and Zaki model [Lundberg J., 2008]:

$$v_{rel} = \alpha_f^{n-1} \tag{2.41}$$

With n being the so called *Richardson Zaki parameter:*

$$n = \begin{cases} 4.65 & Re_{RZ} < 0.2 \\ 4.4Re_{RZ}^{-0.03} & 0.2 > Re_{RZ} < 1 \\ 4.4Re_{RZ}^{-0.1} & 1 > Re_{RZ} < 500 \\ 2.4 & Re_{RZ} > 500, \end{cases} \tag{2.42}$$

where Re_{RZ} is a modified Reynolds number:

$$Re_{RZ} = \frac{Re_p}{v_{rel}}. \tag{2.43}$$

Figure 2.5 shows the iterative algorithm for the calculation of n.

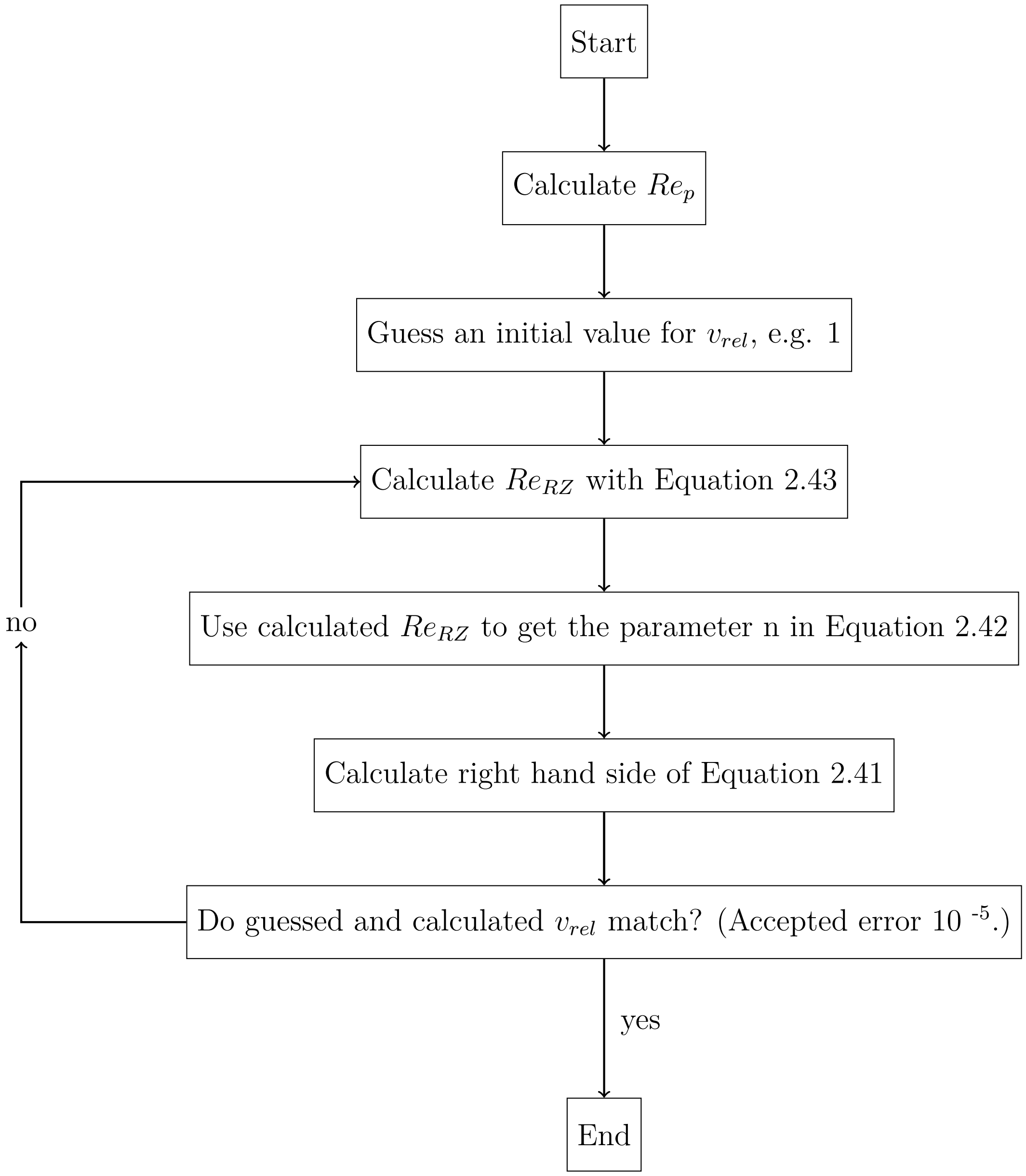


Figure 2.5: Flow chart of the algorithm used to determine v_{rel} in the Richardson and Zaki drag model

2.4.3 Gidaspow

The Gidaspow drag model is a combination of the Wen and Yu drag model, which uses a correlation from the experimental data of Richardson and Zaki, and the Ergun equation [Gidaspow, 1994]. The correlation by Gidaspow is valid if viscous forces are dominating the flow [Lundberg J., 2008].
Equation 2.44 shows the Wen and Yu momentum transfer rate:

$$K_{\text{sf}} = \frac{3}{4} C_{Dspher} \frac{\alpha_f \alpha_s |\vec{u}_s - \vec{u}_f| \rho_f}{d_p} \alpha_f^{-2.65}. \tag{2.44}$$

C_{Dspher} is the drag factor of a spherical particle and related to the Reynolds number by [Rowe et al., 1960]:

$$C_{Dspher} = \begin{cases} \frac{24}{Re_{mean}}\left[1 + 0.15 Re_{mean}^{0.687}\right] & Re_{mean} < 1000 \\ 0.44 & Re_{mean} \geq 1000. \end{cases} \tag{2.45}$$

With Re_{mean} being the so called mean flow Reynolds number:

$$Re_{mean} = \frac{\alpha_f |\vec{u}_f - \vec{u}_s| d_p}{\nu} = Re_p \cdot \alpha_f \tag{2.46}$$

The Ergun equation [Ergun, 1952] was primary derived for packed beds and was later adapted by [Akgiray and Saatci, 1998] for fluidized beds:

$$K_{sf} = A\,\frac{\mu_f (1-\alpha_f)^2}{\alpha_f (d_p \phi)^2} + B\,\frac{\rho_f (|\vec{u}_f - \vec{u}_s|)(1-\alpha_f)}{d_p \phi}, \tag{2.47}$$

with:
ϕ …shape factor (1 corresponds to perfect sphere)
A, B …constants, default: $A = 150$, $B = 1.75$.

The first term on the right hand side (RHS) describes the viscous flow at a low Reynolds number, while the second term describes the kinetic flow at high Reynolds numbers.
The combination of the Wen and Yu drag model from Equation 2.44 and the Ergun equation from Equation 2.47 is considered as the Gidaspow drag model [Gidaspow, 1994]:

$$K_{sf} = \begin{cases} K_{sf}(Ergun) & \alpha_f \leq 0.8 \\ K_{sf}(WenYu) & \alpha_f > 0.8. \end{cases} \tag{2.48}$$

2.4.4 RUC

The Representative Unit Cell (RUC) drag model was first published by [Du Plessis and Masliyah, 1988]. Since then, several modifications have been proposed in literature (see [Lundberg J., 2008]). The underlying principle is geometrical averaging of porous media. This approach ensures the validity for the whole possible range of particle volume fractions. The RUC is of similar form as the Ergun equation (Equation 2.47), with functional expressions for A and B substituting the Ergun model constants (150 and 1.75) [Lundberg J., 2008]:

$$A = \frac{26.8\,\alpha_f^3}{\alpha_s^{\frac{2}{3}}\left(1-\alpha_s^{\frac{1}{2}}\right)\left(1-\alpha_s^{\frac{2}{3}}\right)^2} \tag{2.49}$$

$$B = \frac{\alpha_f^2}{\left(1-\alpha_f^{\frac{2}{3}}\right)^2}. \tag{2.50}$$

2.4.5 Hill Koch Ladd

The Hill-Koch-Ladd (HKL) drag model was first published by [Hill et al., 2001] and modified by [Benyahia et al., 2006] to the version used in this work. The modification extended the validity to a wider range of Reynolds numbers and solid volume fractions. It differs to the previous models, because it is entirely based on computer simulations. The modified HKL model is shown in Equation 2.51:

$$K_{sf} = \frac{3}{4}\,\frac{C_{D,HKL}\alpha_s\alpha_f\rho_f|\vec{u}_f - \vec{u}_s|}{d_p}. \tag{2.51}$$

The drag coefficient $C_{D,HKL}$ is expressed as:

$$C_{D,HKL} = 12\,\frac{\alpha_f^2}{Re_{HKL}}F_{HKL}, \tag{2.52}$$

with F_{HKL} being a dimensionless drag factor which correlates the drag to the particle concentration and the particle radius based Reynolds number:

$$Re_{HKL} = \frac{\rho_f\alpha_f d_p|\vec{u}_f - \vec{u}_s|}{2\,\mu_f}. \tag{2.53}$$

The formulation for Re_{HKL} also explains the coefficient 12 in Equation 2.52 rather than the typical 24, which is related to the particle diameter [Benyahia et al., 2006]. The dimensionless drag factor F_{HKL} is calculated depending on the Reynolds number and the particle volume fraction. The piecewise function for F_{HKL} can be seen in Equation 2.54:

$$F_{HKL} = \begin{cases} 1 + \frac{3}{8}Re_{HKL} & for \quad \alpha_s \leq 0.01 \wedge Re_{HKL} \leq \frac{F2-1}{\frac{3}{8}-F_3} \\ F_0 + F_1 Re_{HKL}^2 & for \quad Re_{HKL} \leq \frac{F_3+\sqrt{F_3^2-4F_1(F_0-F_2)}}{2F_1} \\ F_2 + F_3 Re_{HKL} & \qquad Otherwise, \end{cases} \tag{2.54}$$

with following coefficients:

$$F_0 = \begin{cases} (1-\omega)\;\psi_1 + 10\,\omega\frac{\alpha_s}{(1-\alpha_s)^3} & 0.01 < \quad \alpha_s < 0.4 \\ 10\,\frac{\alpha_s}{(1-\alpha_s)^3} & \qquad \alpha_s \geq 0.4 \end{cases} \tag{2.55}$$

$$F_1 = \begin{cases} \frac{\sqrt{\frac{2}{\alpha_s}}}{40} & 0.01 < \quad \alpha_s \leq 0.1 \\ 0.11 + 0.00051\,e^{11.6\,\alpha_s} & \qquad \alpha_s > 0.1 \end{cases} \tag{2.56}$$

$$F_2 = \begin{cases} (1-\omega)\;\psi_2 + 10\,\omega\frac{\alpha_s}{(1-\alpha_s)^3} & 0.01 < \quad \alpha_s < 0.4 \\ 10\,\frac{\alpha_s}{(1-\alpha_s)^3} & \qquad \alpha_s \geq 0.4 \end{cases} \tag{2.57}$$

$$F_3 = \begin{cases} 0.9351\alpha_s + 0.03667 & \alpha_s < 0.0953 \\ 0.0673 + 0.212\alpha_s + \frac{0.0232}{(1-\alpha_s)^5} & \alpha_s \geq 0.0953. \end{cases} \tag{2.58}$$

With:

$$\psi_1 = \frac{1 + 3\sqrt{\frac{\alpha_s}{2}} + (135/64)\alpha_s ln(\alpha_s) + 17.14\alpha_s}{1 + 0.681\alpha_s - 8.48\alpha_s^2 + 8.16\alpha_s^3} \tag{2.59}$$

$$\psi_2 = \frac{1 + 3\sqrt{\frac{\alpha_s}{2}} + (135/64)\alpha_s ln(\alpha_s) + 17.89\alpha_s}{1 + 0.681\alpha_s - 11.03\alpha_s^2 + 15.41\alpha_s^3} \tag{2.60}$$

$$\omega = e^{-\frac{10(0.4-\alpha_s)}{\alpha_s}}. \tag{2.61}$$

2.4.6 Tenneti

The drag model developed by [Tenneti et al., 2011] is based on particle resolved direct numerical simulation (DNS) through a method called Particle-resolved Uncontaminated - fluid Reconcilable Immersed Boundary Method (PUReIBM). That means, that the Navier-Stokes equations are solved with no-slip and no-penetration boundary conditions using an immersed boundary forcing term at the particle surfaces. Further information on the PUReIBM can be found elsewhere [Tenneti et al., 2011].
The derived expression for the dimensionless drag is:

$$F_{Ten} = \frac{F_{isol}(Re_{mean})}{(1-\alpha_s)^3} + F_4 + F_5. \tag{2.62}$$

With F_{isol} being the drag force acting on an isolated sphere moving in an unbounded medium. F_{isol} is obtained from the correlation by [Shiller and Naumann, 1935]:

$$F_{isol} = \begin{cases} 1 + 0,15\ Re_p^{0.687} & Re_p < 1000 \\ \frac{0.44\ Re_p}{24} & Re_p > 1000 \end{cases} \tag{2.63}$$

$$F_4 = \frac{5.81\ \alpha_s}{(1-\alpha_s)^3} + 0.48\ \frac{\alpha_s^{\frac{1}{3}}}{(1-\alpha_s)^4} \tag{2.64}$$

$$F_5 = \alpha_s^3\ Re_{mean}\left(0.95 + \frac{0.61\ \alpha_s^3}{(1-\alpha_s)^2}\right) \tag{2.65}$$

and Re_{mean} being the so called mean flow Reynolds number defined in Equation 2.46.

The momentum exchange rate K_{sf} is calculated by the expression of Levenspiel:

$$K_{sf} = 0.75 \cdot \frac{\rho\ \alpha_s\ \alpha_f\ |\vec{u}_f - \vec{u}_s|}{d_p} \cdot C_{D,Ten}, \tag{2.66}$$

where:

$$C_{D,Ten} = F_{isol} \cdot \alpha_f + \frac{24 \cdot \alpha_f^2}{Re_{mean}} \cdot (F_4 + F_5). \tag{2.67}$$

2.4.7 Beetstra

[Beetstra et al., 2007] developed a drag model for fluid flow past monodisperse arrays of spheres with simulations based on the Lattice Boltzmann method (LBM). The LBM is a class of CFD methods solving non-equilibrium thermodynamical systems using the Boltzmann equation[1] on a discretized phase space [Lerner and Trigg, 1991]. [Beetstra et al., 2007] derived following correlation for the dimensionless drag force for monodisperse systems as a function of Re_{mean} and α_s from the simulations:

$$\begin{aligned} F_{Beet} = \frac{10\alpha_s}{(1-\alpha_s)^2} + (1-\alpha_s)^2\,(1+1.5\alpha_s^{\frac{1}{2}}) + \frac{0.413 Re_{mean}}{24(1-\alpha_s)^2} \cdot \\ \frac{(1-\alpha_s)^{-1} + 3\alpha_s(1-\alpha_s) + 8.4 Re_{mean}^{-0.343}}{1 + 10^{3\alpha_s} Re_{mean}^{\frac{-(1+4\alpha_s)}{2}}} \end{aligned} \tag{2.68}$$

The calculation of K_{sf} is similar to the one in subsection 2.4.6:

$$K_{sf} = 0.75 \frac{\rho\ \alpha_s\ \alpha_f\ |\vec{u}_f - \vec{u}_s|}{d_p} C_{D,Beet}, \tag{2.69}$$

where:

$$C_{D,Beet} = \frac{24\ \alpha_f}{Re_{mean}} F_{Beet}. \tag{2.70}$$

---

[1] general form: $\frac{df}{dt} = \left(\frac{\partial f}{\partial t}\right)_{force} + \left(\frac{\partial f}{\partial t}\right)_{diff} + \left(\frac{\partial f}{\partial t}\right)_{coll}$ where the $force$ term corresponds to the forces exerted on the particles by an external influence, the $diff$ term represents the diffusion of particles, and $coll$ is the collision term accounting for the forces acting between particles in collisions [McGraw-Hill, 1994].

2.5 Other Effects in Multiphase Flows

2.5.1 Added Mass Force

Added mass or virtual mass is the mass added to a system because an accelerating or decelerating body must move some volume of the surrounding fluid as it moves through it. This is a common issue because the object and the surrounding fluid cannot occupy the same space at the same time. As a result, moving particles seem to have a larger mass than their actual mass [Jakobsen, 2008].

For a spherical particle submerged in an inviscid, incompressible fluid, the virtual mass force is:

$$\vec{F}_{virt} = \frac{\rho_f \cdot V_p}{2} \cdot \left(\frac{D\vec{u}_f}{Dt} - \frac{d\vec{u}_s}{dt} \right), \tag{2.71}$$

with:
V_p ... volume of the particle
$\frac{D}{Dt}$ ... material derivative.

While taking a look at the momentum equation for the particle, the origin of the notion "virtual mass" becomes clear:

$$m_p \cdot \frac{d\vec{u}_s}{dt} = \sum \vec{F} + \frac{\rho_f \cdot V_p}{2} \left(\frac{D\vec{u}_f}{Dt} - \frac{d\vec{u}_s}{dt} \right), \tag{2.72}$$

where $\sum \vec{F}$ is the sum of all other force terms on the particle, such as gravity, pressure gradient, drag, lift, Basset force, etc.
Moving the derivative of the particle velocity from the RHS of the equation to the left hand side (LHS) we get:

$$\left(m_p + \frac{\rho_f \cdot V_p}{2} \right) \frac{d\vec{u}_s}{dt} = \sum \vec{F} + \frac{\rho_f \cdot V_p}{2} \frac{D\vec{u}_f}{Dt}, \tag{2.73}$$

so the particle is accelerated as if it had an added mass of half the fluid it displaces, and there is also an additional force contribution on the RHS due to acceleration of the fluid [Crowe et al., 1998].

2.5.2 Mass Inertia

The particle response time describes the time it takes for a particle to respond to a change of the velocity of the fluid phase and can be defined as [Brennen, 2005]:

$$\tau_{res} = \frac{\rho_s d_p^2}{18\mu_f}. \tag{2.74}$$

It is used in the Stokes number (St), which is a dimensionless number that describes the behavior of a particle suspended in fluid flow. It is defined as the ratio between the particle response time and the characteristic time of the fluid:

$$St = \frac{\tau_{res}}{\tau_{char}}, \tag{2.75}$$

with:

$$\tau_{char} = \frac{L}{u_f}, \tag{2.76}$$

where L is the characteristic length.
At small Stokes numbers, particles follow changes in the fluid flow closely, whereas at high Stokes numbers the particles will detach from flow according to mass inertia.

2.5.3 Turbulent Dispersion

In fluid dynamics, turbulence or turbulent flow is any pattern of fluid motion characterized by chaotic changes in pressure and flow velocity [Batchelor, 2000].
Turbulent flows depend in many ways on time and location. Typical characteristics are high Reynolds numbers and a varying velocity field. To describe this type of flow, statistical methods are used. An important class of these methods are so called turbulence models, which approximate the mean flow [Kuhlmann, 2007].
In multiphase flows, turbulence in the continuous phase can have an effect on the particles in the dispersed phase and cause them to be transported from regions of high concentration to regions of low concentration. This affection of the migration of the particles is referred to as *turbulent dispersion* [Burns et al., 2004].
A so called turbulent dispersion force is introduced, which describes the effects of the interphase turbulent momentum transfer and acts like a turbulent diffusion in dispersed flows. For a dispersed (solid) phase s and a continuous phase f, the turbulent dispersion force can be modeled as [Bauer et al., 2016]:

$$K_{sf}\left(\widetilde{\vec{v}}_s - \widetilde{\vec{v}}_f\right) = K_{sf}\left(\vec{v}_s - \vec{v}_f\right) - K_{sf}\vec{v}_{dr}. \tag{2.77}$$

With the term on the LHS representing the instantaneous drag, the first term on the RHS $K_{pq}\left(\vec{v}_p - \vec{v}_q\right)$ describing the mean momentum exchange between the two phases. The second term is $K_{sf}\vec{v}_{dr}$ describing the turbulent dispersion force:

$$\vec{F}_{td,f} = -\vec{F}_{td,s} = -f_{td,lim} K_{sf}\vec{v}_{dr}, \tag{2.78}$$

with $\vec{v}_{dr}$ being the drift velocity. It is caused by particle transport due to turbulent fluid motion. $f_{td,lim}$ is a factor used to limit the turbulent dispersion force.

3 Methods and Preliminary Evaluations

3.1 OpenFOAM Implementation

All simulations were made with the open source CFD software Open source Field Operation and Manipulation (OpenFOAM)[1] [Weller et al., 1998]. OpenFOAM is a C++ based toolbox for solving continuum mechanics problems. There is a variety of already existing solvers for different problems including multiphase flow, electromagnetics, combustion, and heat transfer. Additional solvers can be added to the toolbox by the user.

3.1.1 DPMFoam and MPPICFoam

DPMFoam and MPPICFoam are both Lagrangian solvers for particle-laden flow. They were developed for the transient calculation of kinematic particle clouds including the effect of the volume fraction of particles on the continuous phase. MPPICFoam uses the Multiphase Particle-In-Cell method (MP-PIC) described in subsubsection 2.2.3.2 to represent collisions without resolving particle-particle interactions [Weller et al., 2018].

In DPMFoam, the PISO-SIMPLE (PIMPLE) algorithm is used to solve the Navier-Stokes equations [Moukalled et al., 2015]. PIMPLE is a combination of Pressure Implicit with Splitting of Operator (PISO) and Semi-Implicit Method for Pressure-Linked Equations (SIMPLE). Both are iterative solvers with the difference, that SIMPLE is a steady-state algorithm, while PIMPLE is a transient one. An easy way to think about PIMPLE, is to imagine it as a SIMPLE algorithm done for every time-step. (The SIMPLE workflow is shown in Figure 3.1). One time-step is completed, when the criteria for time step convergence (default 10^{-5}) is reached. The solver will continue with the next time-step. When solving the Navier-Stokes equations, the solver is taking a guessed value for pressure p^* that fits the velocity

---

[1] OPENFOAM® is a registered trademark of OpenCFD Ltd licensed in perpetuity to the OpenFOAM Foundation.

term. When the pressure field is updated, this guessed value is used:

$$p^{k+1} = p^k + p^* \tag{3.1}$$

The pressure term is always under-relaxated, otherwise the computed value for the pressure correction would be too high [Bunge, 2003]. After the velocity field is updated, the received pressure value is set as the new guessed pressure value $p = p^+$ and the procedure repeats until convergence [Moukalled et al., 2015].
Figure 3.1 shows the iterative SIMPLE algorithm.

3.1.2 Submodels implemented in MPPICFoam

This section gives a short overview about the submodels implemented in MPPICFoam and how they work. The parameters for these models are set in the `kinematicCloudProperties` file inside the `constant` directory of an OpenFOAM case.

Table 3.1: Explanation of the submodels implemented in the MPPICFoam solver.

Submodel	**Explanation**
`packingModel`	There are two options: implicit and explicit. For implicit, the time evolution of the solid volume fraction is solved implicitly on the Eulerian mesh. The computed flux is then applied to the Lagrangian field. For explicit, the inter-particle stress is calculated using the current particle position. The stress force is then applied only to particles that are moving towards regions of close pack.
`dampingModel`	If the damping model `relaxation` is chosen, the calculation for the damping time τ_D will be according to [O'Rourke and Snider, 2010].
`isotropyModel`	For this work, the isotropy model `stochastic` is chosen, meaning that the return-to-isotropy model of [O'Rourke and Snider, 2012] is employed. For that purpose, a Gaussian-plus-delta distribution chooses some velocities of certain particles to be modified. The modification depends on a time-scale model, which randomizes the velocities. The lower the value of the time-scale, the greater the affection of the particle velocities.
`injection model`	Handles the injection of the dispersed phase. A description can be found in section 3.4.
`particleStressModel`	Describes the inter-particle stress. The mode chosen for this work was the one by [Harris and Crighton, 1994].
`patchInteractionModel`	Handles the interaction between a particle and a wall. All patches defined in the geometry have to be set separately. Possible settings are `stick`, `rebound`, and `escape`, which are self-explanatory. In this work, the option `rebound` is chosen. The rebound is set with the so-called coefficient of restitution e.

Under-relaxation Under-relaxation can help to keep a simulation stable. It reduces solution oscillations. After each iteration, at each cell, a new value for a variable X in cell i is updated using following equation [Versteeg and Malalasekera, 1995]:

$$X_i^{new,used} = X_i^{old} + \alpha(X_i^{new,predicted} - X_i^{old}), \tag{3.2}$$

where α is the under-relaxation factor. $\alpha = 1$ meaning no under-relaxation. In OpenFOAM, the option for `underrelaxation` is found in the `fvSolution` file inside the `system` directory. An entry looks like this:

```
relaxationFactors
{
    fields
    {
    }
    equation
    {
        U.air        0.8;
        p            0.8;
        alpha.air    0.8;
    }
}
```

Listing 3.1: Settings for the underrelaxation of the parameters air velocity, pressure, and fluid volume fraction in OpenFOAM.

Coefficient of restitution The coefficient of restitution e was first described by Isaac Newton [Weir and McGavin, 2008] and defines the ratio of the final to initial relative velocity between two objects after they collide. In CFD simulations, it defines the particle-particle and particle-wall interactions by following formula:

$$e = \frac{relative\ velocity\ after\ collision}{relative\ velocity\ before\ collision}, \tag{3.3}$$

with $e = 1$ meaning that the velocity magnitude of the particle after collision is the same as before.

In the OpenFOAM implementation, the rebound is separated into normal and tangential velocity and e describes the normal velocity and μ the tangential one. For $\mu = 0$: $u_{t,new} = u_{t,old}$

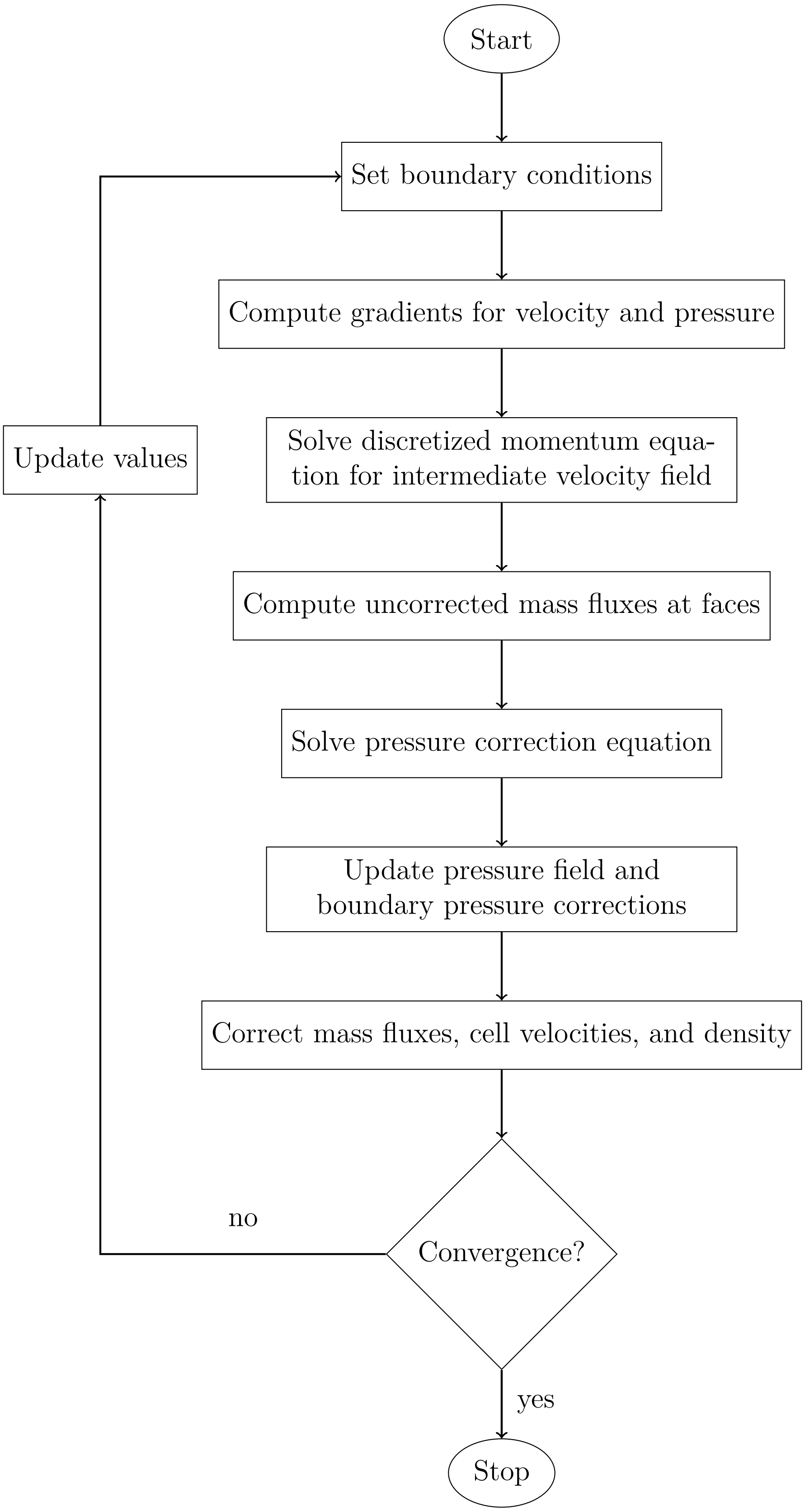


Figure 3.1: Flow chart of the basic steps in the solution update of the SIMPLE algorithm [Moukalled et al., 2015]

3.2 Comparison of the Drag Models

For the comparison of the drag models presented in section 2.3, the drag force K_{sf} is calculated for each model in a defined test case. The results are shown in the following figures as a function of Reynolds number and various solid volume fractions α_s. The physical properties employed for the comparison are presented in Table 3.2:

Table 3.2: Physical properties employed for the comparison of different drag models.

Property	Unit	Value
Temperature	K	1200
Pressure	Pa	10^5
Kinematic viscosity	$\frac{m^2}{s}$	$16.68 \cdot 10^{-5}$
Dynamic viscosity	$Pa \cdot s$	$4.84357 \cdot 10^{-5}$
Particle diameter	m	0.02
Relative flow velocity	$\frac{m}{s}$	$0.0001 - 200$
Density	$\frac{kg}{m^3}$	0.29035
Shape factor	-	1 (perfect sphere)
Solid volume fraction	-	$0.1 - 0.72$

The different drag models use varying definitions of the Reynolds number (compare subsection 2.4.1 to 2.4.7). Table 3.3 summarizes the different definitions for the sake of clarity:

Table 3.3: Comparison of Reynolds numbers used in different drag models.

Drag Model	Reynolds number
Syamlal O'Brien	$Re_p = \frac{\rho_f d_p \lvert \vec{u}_s - \vec{u}_f \rvert}{\mu_f}$
Richardson Zaki	$Re_{RZ} = \frac{Re_p}{v_{rel}}$
Gidaspow	$Re_{mean} = Re_p \cdot \alpha_f$
Tenneti	Re_{mean}
HKL	$Re_{HKL} = \frac{\rho_f \alpha_f d_p \lvert \vec{u}_f - \vec{u}_s \rvert}{2 \mu_f} = \frac{Re_{mean}}{2}$
Beetstra	Re_{mean}
RUC	Re_p

Figure 3.2 shows the results for the calculation of K_{sf} for the defined test case.

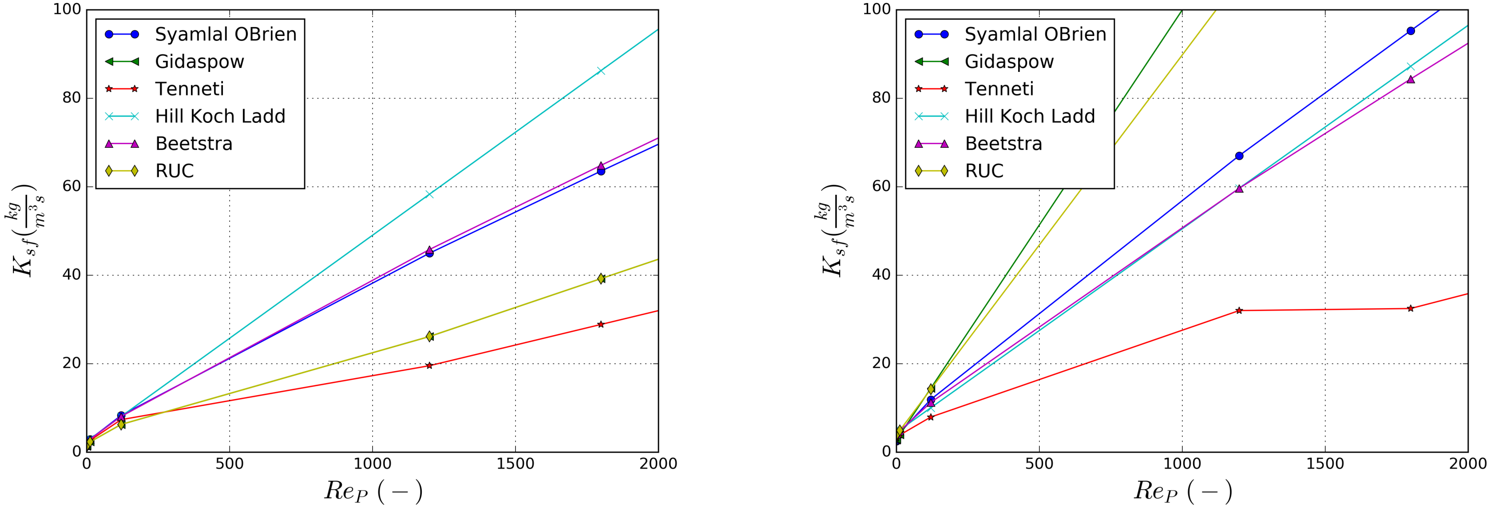
It can be noted, that for the most dilute system($\alpha_s = 0.1$) (Figure 3.2a) the models agree best. HKL predicts the highest values for the drag force. The Syamlal O'Brien and Beetstra model agree very well, which is worth mentioning, because the calculation of the drag force is quite different for these two models. The RUC and Tenneti model predict the lowest values.

For the slightly denser system in Figure 3.2b three groups of predictions can be seen: Gidaspow and RUC predict the highest values for K_{sf}. At this solid volume fraction ($\alpha = 0.25$), the difference between these two models is only in the definitions of the parameters A and B, since they both use the Ergun equation for the prediction of the drag force. Syamlal O'Brien, HKL, and Beetstra model agree well, but predict lower values than the two models mentioned before, and the Tenneti model again predicts the lowest values.

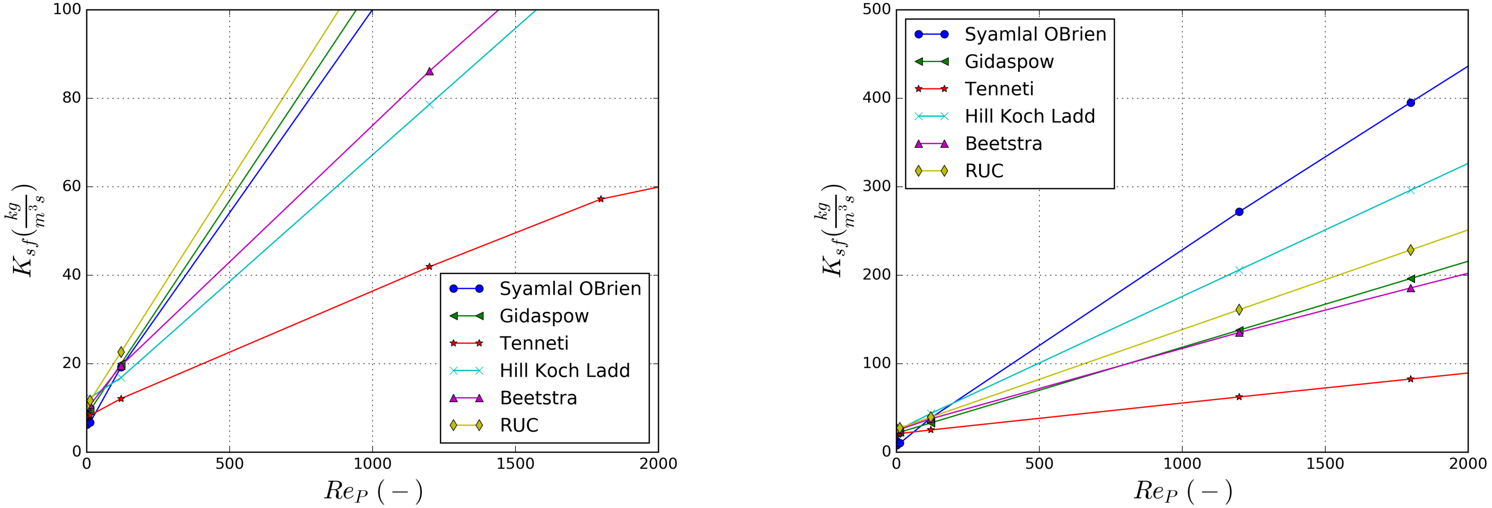
The results for $\alpha_s = 0.5$ are very similar: Gidaspow, RUC, and Syamlal O'Brien predict higher values for K_{sf}, HKL and Beetstra predict slightly lower values, and Tenneti predicts significant lower values.

The plot for the calculation with a solid volume fraction of 0.72 shows a bit of a difference: The Syamlal O'Brien model predicts the highest K_{sf} values, followed by the HKL model. The RUC, Gidaspow, and Beetstra model agree pretty well and the Tenneti model predicts the lowest values once again.

Since the Tenneti model always predicts the lowest values for the drag force, it is doubtful that this model is suitable for a calculation with the parameters used in this case. The same assumption is made for the Richardson Zaki models, which does not predict any reasonable values at all and is therefore not visible in Figure 3.2.



(a) $\alpha_s = 0.1$ (b) $\alpha_s = 0.25$



(c) $\alpha_s = 0.5$ (d) $\alpha_s = 0.72$

Figure 3.2: Comparison of different drag models for a defined test case. Drag force (K_{sf}) as a function of Reynolds number (Re_p).

3.3 Evaluation of the Drag Models

For a better and easier selection of a suitable drag model, an additional evaluation was done. The aim of this task was to model an existing test case and thereby evaluate the drag models through comparing to the experimental results.
The test case of choice is a random packed bed, described by [Beetstra et al., 2007]. The experimental pressure drop results of the monodisperse system were used for the evaluation. The original test setup consisted of a cylindrical column with an inner diameter $D = 80\ mm$ and length $L = 300\ mm$ placed in a jacket which was used for temperature regulation. The column was filled with monodisperse random fillings of glass spheres, at five different diameters: 1.0, 1.5, 2.0, 2.5 and 4.0 mm. A gear pump was used to circulate liquid through the system with an adjustable pump frequency. For the experiment with low Reynolds numbers, glycerol was used and for the ones with higher Reynolds numbers water was used. The pressure drop was measured with a pressure differential sensor that was connected to four tubes in the wall. Figure 3.3 shows a schematic representation of the experimental setup used by [Beetstra et al., 2007].

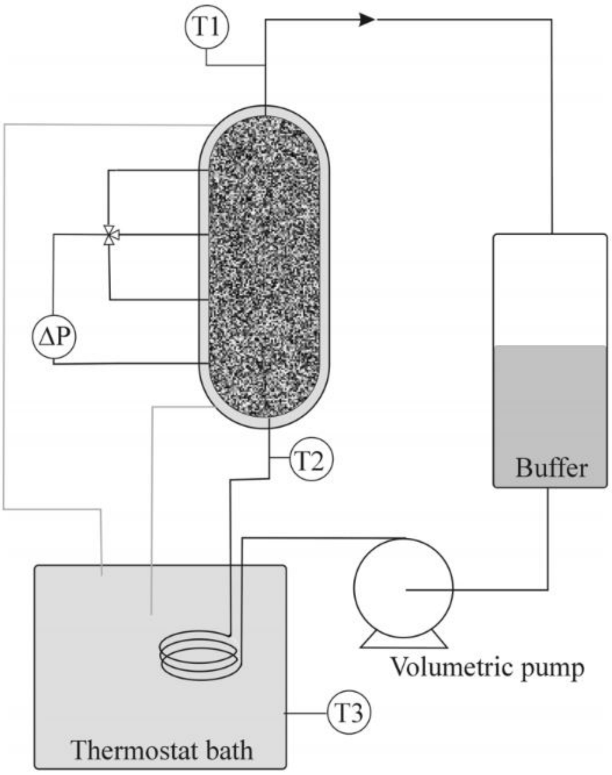


Figure 3.3: Schematic representation of the experimental setup performed by [Beetstra et al., 2007].

Figure 3.4 shows the experimental results obtained by [Beetstra et al., 2007].

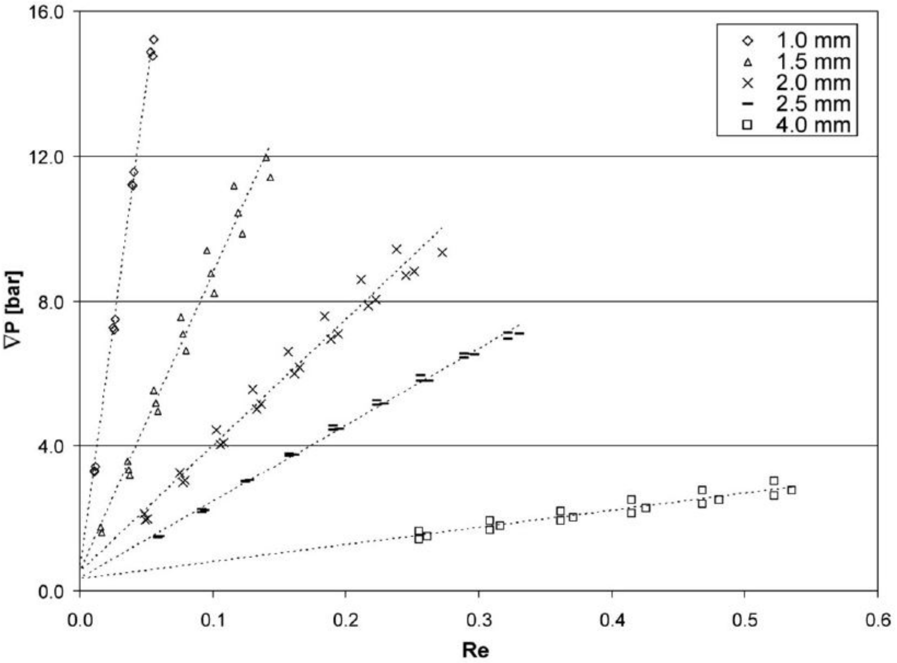


Figure 3.4: Measured pressure drops for glass spheres of five different diameters, [Beetstra et al., 2007].

The validation was done with a python[2] script. To get the pressure drop directly from the K_{sf} calculation, following expression was used:

$$\Delta p = K_{sf} \cdot U \cdot L. \tag{3.4}$$

The velocity of the liquid was determined via the experimental Reynolds number, which was extracted from Figure 3.4 by following formula:

$$u_f = \frac{Re\ \nu}{L}, \tag{3.5}$$

where:
ν ... kinematic viscosity
L ... length of the column.

Figure 3.5 compares the experimental results of [Beetstra et al., 2007] and the calculated pressure losses.

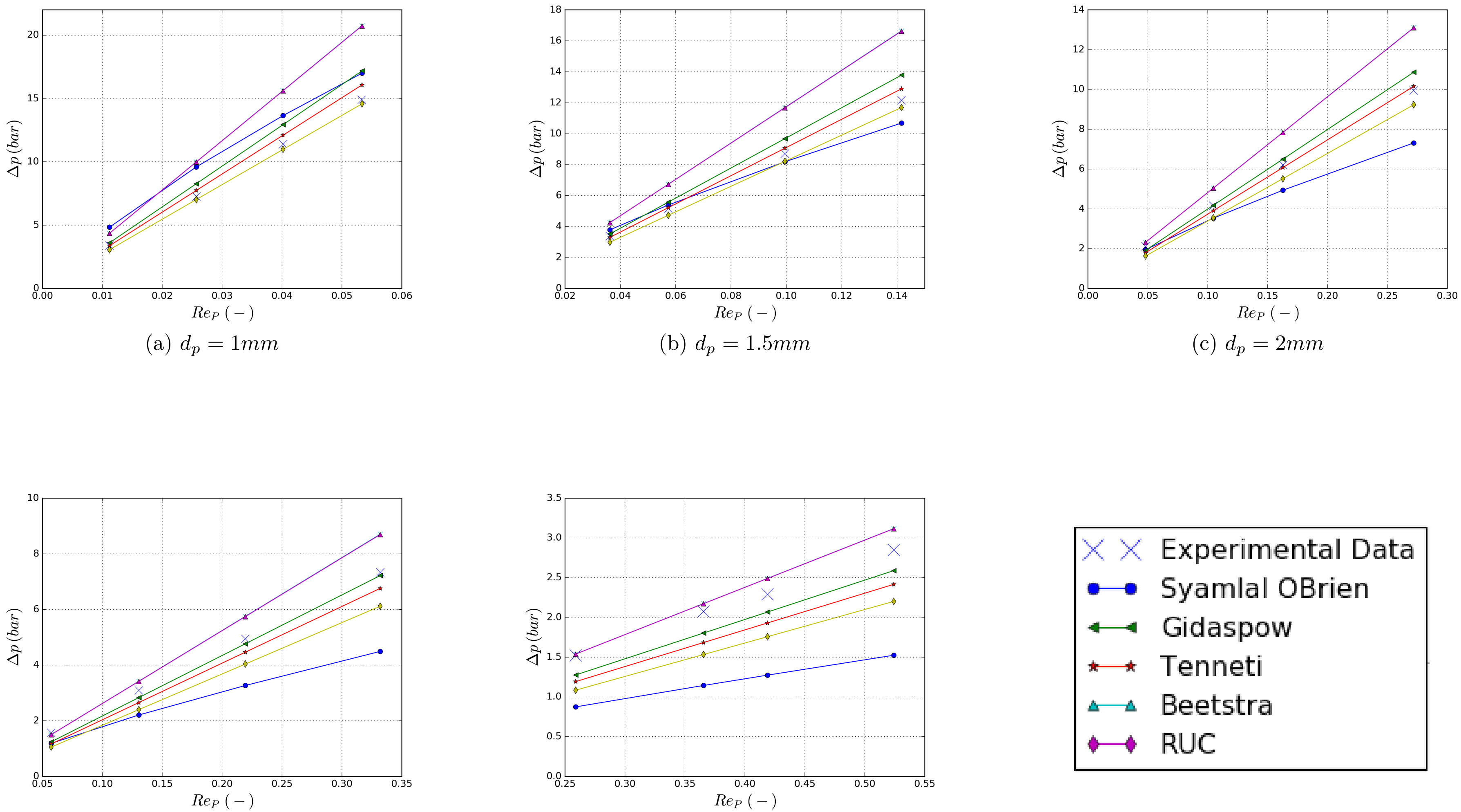
In general, all the drag models except the Richardson Zaki model predict reasonable values for the pressure drop at all sphere sizes. Even though the implementation of this model was checked several times, no promising results were made and, therefore, this model was not considered in the subsequent work.

---

[2] Python is an interpreted high-level programming language for general-purpose programming, that was created by Guido von Rossum. All Python releases are Open Source and are managed by the non-profit Python Software Foundation.

The Tenneti model agrees very well with the experimental data, being the model which predicts the closest values for the test case with 1.0, 1.5, and 2.0 mm glass spheres and the second closest for the 2.5 and 4.0 mm spheres.

The RUC and Gidaspow model also predict reasonable results for the cases with 1.0 to 2.5 mm glass spheres. Syamlal O'Brien's model predicts higher values for 1.0 mm and lower values for the other cases than measured, while the Beetstra model over-predicts the pressure drop for all cases except for the 4.0 mm spheres, where it agrees best with the experimental results. Generally, calculated values deviate most for the 4.0 mm sphere case.



(a) $d_p = 1mm$

(b) $d_p = 1.5mm$

(c) $d_p = 2mm$

(d) $d_p = 2.5mm$

(e) $d_p = 4mm$

(f) legend

Figure 3.5: Comparison of the calculated pressure drop in a random close packed bed of different drag models with the experimental data of [Beetstra et al., 2007].

3.4 Evaluation of DPMFoam Simulations

The *DPMFoam* solver is evaluated employing the experimental data of [Haddadi et al., 2017]. They investigated the influence of different particle types on the packing characteristics. Therefore, they made experiments and simulations of the filling process of random packed beds. This work reproduces the filling process with the DPMFoam solver and compares the characteristics of the packings with experimental and DEM simulation results.

[Haddadi et al., 2017] experimented with cylinders randomly filled with monodisperse spheres and simulated the same setup using discrete elements method (DEM). This case is employed for the evaluation of the DPMFoam solver and the results were compared to the experimental and simulation results of [Haddadi et al., 2017].

The test case was a cylinder of 0.02 m diameter filled with 0.006 m spheres. The packing has not been compacted.

The spheres were falling freely into the cylinder, which was standing in a fully upright position. Once the filling reached a height of 0.13 m, the simulation was stopped and the filling height was corrected by keeping only those particles that were completely underneath the height of 0.13 m.

A blockstructured cylindrical mesh was created in blockMesh (with support of a python script) for the solver validation. The base can be seen in Figure 3.6. The total domain height was 0.15 m.

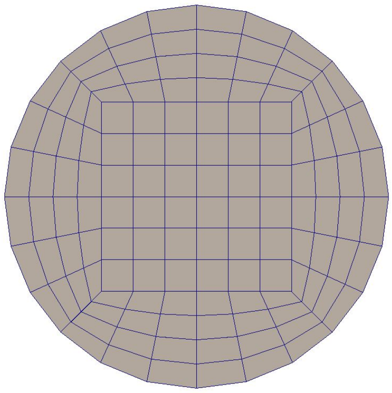


Figure 3.6: Cylindrical base mesh of the fluidized bed domain.

For the filling of the particles, an injection model with following settings was used in DPMFoam. Table 3.4 explains the occuring keywords.

```
injectionModels
{
    model1
    {
        type                patchInjection;
        patchName           inlet;
        duration            6;
        parcelsPerSecond    100;
        massTotal           0;
        parcelBasisType     fixed;
        nParticle           1;
        SOI                 0;
        U0                  (0 0 0);
        flowRateProfile     constant 1;
        sizeDistribution
        {
            type            fixedValue;
            fixedValueDistribution
            {
                value     0.006;
            }
        }
    }
}
```

Listing 3.2: Settings for the injection model for the DPMFoam simulation of a random packed bed of mono-disperse spheres

Table 3.4: Explanation of the keywords used in the Lagrangian settings of the DPMFoam solver.

Parameter	**Explanation**
`patchName`	defines where the particles enter the geometry
`duration`	defines how long the injection lasts
`parcelsPerSecond`	injected parcels per second
`massTotal`	total injected mass (not used when option "fixed" is selected in the following)
`parcelBasisType`	fixed, mass and number can be selected; fixed sets the number of total particles by muliplying nParticle, parcelsPerSecond and duration
`nParticle`	particles per parcel
`SOI`	start of injection
`U0`	start velocity of the parcles
`flowRateProfile`	constant means a constant flow of parcels
`sizeDistribution`	fixed value means no distribution, all particles have the same diameter (0.006 m)

The total number of injected particles is calculated by:

$$1\ \frac{particle}{parcel} \cdot 100\ \frac{parcels}{s} \cdot 6.5\ s = 650\ particles \tag{3.6}$$

In this simulation, the principle of summing up single particles to a parcel for lowering computational cost is not used, so every particle is used in the simulation. Therefore the parameter *nParticle* is set to 1.
Figure 3.7 shows the packed bed before and after the height correction.

Table 3.5: Comparison between experimental measurements, DEM simulations, and DPMFoam simulations.

	Reality	DEM	DPMFoam
Number of Particles	525	533	516

Table 3.5 shows the results of the particle counting process. The deviation of the DPMFoam results to the experiment is $\Delta = 1.9\%$ and $\Delta = 3.2\%$ to the DEM simulation results. Figure 3.8 indicates, that particles follow a certain pattern in the filling process by showing their centres of mass in the horizontal plane. Big fluctuations of the porosity are visible: Lots of particles are located directly at the wall, where the particles form a A-B-A style packing, whereas in the center part of the filling two denser and two more dilute regions are formed. The DPMFoam and DEM results of [Haddadi et al., 2017] show good agreement.

Figure 3.7: Creation of a random packed bed with mono-dispersed spheres: a - filled bed, b - corrected bed height 0.13 m.

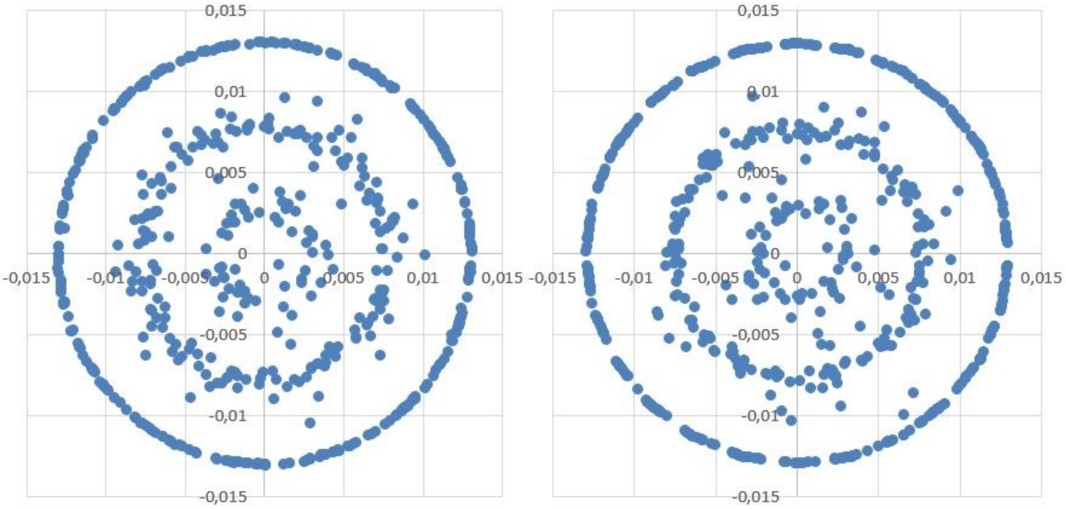


Figure 3.8: Centres of mass of the particles mapped on top view in m. Left: DEM, right: DPMFoam. DEM results from [Haddadi et al., 2017].

3.5 Evaluation of MPPICFoam Simulations

The MPPICFoam solver is evaluated with three different test cases. The first includes the determination of the minimum fluidization velocity of a packed bed. The second and third evaluation cases are cold flow models of blast furnaces.

3.5.1 Fluidized Bed Case

[Jayarathna et al., 2017] measured and determined the minimum fluidization velocity and the pressure drop of Geldart type A particles [Geldart, 1973] using CFD. Table 3.6 shows the properties of the employed particles. For the experiments and simulations, they filled a 1.2 m high cylinder having a 0.084 m diameter up to a height of 0.17 m. After the filling process, they transferred the particle positions to a different case setup to perform the fluidization simulations. During the fluidization experiments, they monitored the pressure 2.5 and 12.5 cm above the distributor plate at the bottom of the cylinder and increased the flow rates sequentially after a 30 s qualifying period.

Table 3.6: Properties of the zirconia particles used in the experiments of [Jayarathna et al., 2017]

Property	**Unit**	**Value**
Material	-	Zirconia
Skeletal density	kg/m^3	3830
Bulk density	kg/m^3	2270
Particle size range	μm	45-100
D_{50}	μm	70
Porosity	-	0.6
Sphericity	-	0.95

The simulation was run for 30 seconds for each flow rate and then increased to the next value.

[Jayarathna et al., 2017] employed the CFD software Barracuda[3] for their simulations with a 25000 cell mesh. They performed all calculations using the Gidaspow drag model, but they adjusted the A value to adapt their simulations to the experimental data.

For the purpose of evaluating the MPPICFoam solver with this experiment, a cylindrical blockMesh was made with identical dimensions and filled in a similar way as described in section 3.4. During the filling process, the option `coupled` was turned

---

[3] Barracuda® is a registered trademark of CPFD Software LLC.

off, meaning that the solid phase is not affecting the fluid phase. This was done to reduce the computational time. The number of parcels injected for reaching the filling height of 17 cm was calculated with following formula:

$$n_P = \frac{V_{cyl}}{V_p}\alpha_s \tag{3.7}$$

With V_{cyl} being the volume of the cylinder until the filling height:

$$V_{cyl} = \frac{\pi}{6}d_p^3 \tag{3.8}$$

Table 3.7 displays the used values for the filling process:

Table 3.7: Simulation settings for the fluidized bed test case.

Parameter	**Unit**	**Value**
d_{cyl}	m	0.084
h_{cyl}	m	0.17
V_{cyl}	m^3	$9.42 \cdot 10^{-4}$
d_p	m	$7 \cdot 10^{-5}$
V_p	m^3	$1.79594 \cdot 10^{-13}$
α_p	-	0.6
Number of Particles	-	3 147 338 404
nParticle (pro Parcel)	-	5000
Total Parcels	-	629468
Duration	s	20
parcelsPerSecond	-	314734

After the filling process had been completed, the file `kinematicCloudPositions`, which includes the positions of the particles as Cartesian coordinates, was copied into the fluidization case setups to provide an already filled cylinder for the fluidized bed simulations. For each one of the six working drag models and each superficial air velocity (0.011, 0.013, 0.015, 0.016, 0.018, and 0.02 m/s) a simulation was set up, making it a total of 36 simulations. Apart from the drag models and the superficial air velocity, all settings were the same.

The steady air flow was feeded at the bottom patch for 30 seconds. After the simulation had been completed, the first 4 time-steps were moved into a sub-directory, because a steady air flow was not guaranteed in this early phase of the simulation. In paraview, an average value for p and α_f was calculated for the remaining 26 time-steps and the pressure drop between 2.5 and 12.5 cm above the bottom wall was calculated.

3.5.2 Raceway Simulations

For the evaluation of the capability of the MPPICFoam solver to predict a raceway formation, two simulations were made. At first, the work of [Mojamdar et al., 2018] was investigated (subsubsection 3.5.2.1). Early attempts showed, that the solver had problems with the fine mesh in the tuyere region. Therefore, a second evaluation with a simpler mesh (subsubsection 3.5.2.2) was done.

3.5.2.1 Raceway Simulation 1

The first raceway evaluation was done by reproducing the experimental work of [Mojamdar et al., 2018]. They investigated the raceway formation in a 2D moving bed. In their work, an experimental setup was constructed to replicate a moving bed, where the size of the developing raceway was investigated concerning different parameters such as gas velocity, bed height, and particle flow rate. According to [Burgess, 1985], a raceway is formed, when air is introduced horizontally into the packed bed of particles.

[Mojamdar et al., 2018] assembled a two dimensional apparatus, made of acrylic glass (poly-methyl methacrylate) and having the dimensions of 500 x 1000 x 60 mm. It is shown in Figure 3.9.

They used polyethylene particles with following properties for the experiments:

Table 3.8: Properties of the polyethylene particles used in the evaluation experiments of [Mojamdar et al., 2018].

Property	**Unit**	**Value**
Material	-	polyethylene
Skeletal density	kg/m^3	940
Shape	-	cylindrical
Length to diameter ratio	-	1.1
Equivalent diameter	mm	4
Mean bed porosity	-	0.398

The formation of the raceway was observed at four different filling heights above the tuyere: 450, 550, 650 and 800 mm. For each experiment, they blew air through the tuyere with various velocities up to 0.6 m/s.

For the validation of this experiment, a similar geometry was drawn in Gambit[4]. Since OpenFOAM can only handle 3D simulations, the depth of the mesh was set to one cell to simulate a 2D mesh. However, OpenFOAM does not solve for the

---

[4] Gambit® is a software package for building mesh models for CFD and developed by ANSYS, Inc.

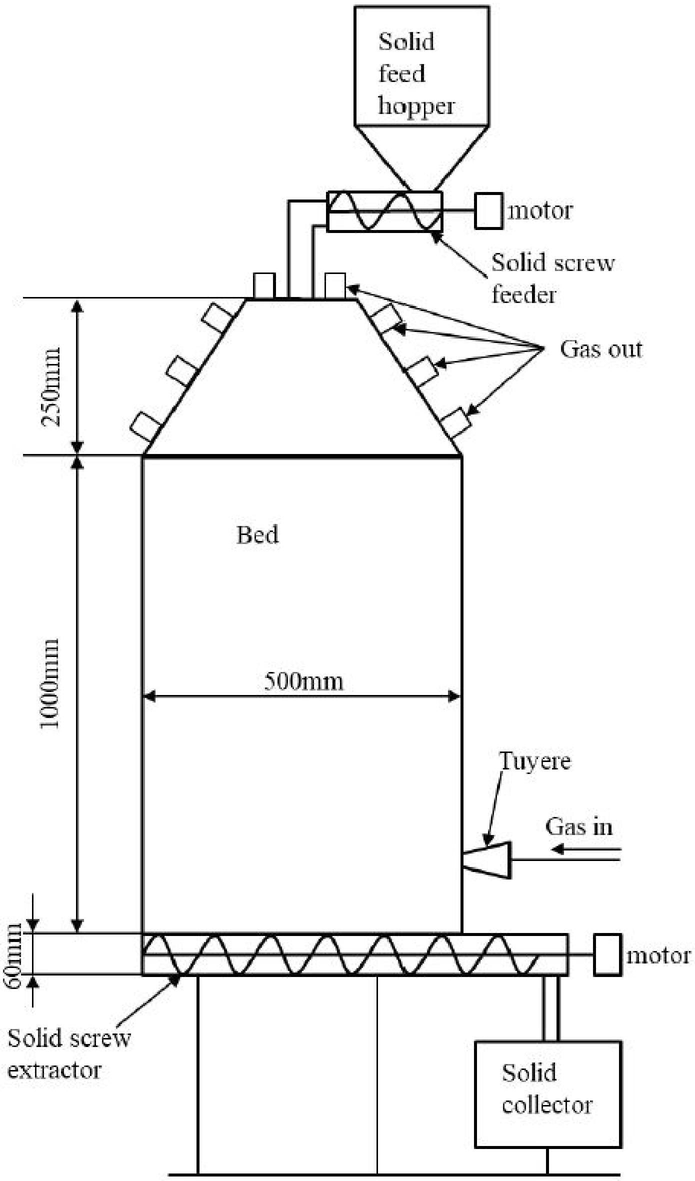


Figure 3.9: Schematic diagram of the experimental setup of [Mojamdar et al., 2018] for the raceway formation in a moving bed.

third dimension. The width and height were set like in the original experiment to 500 *mm* and 1000 *mm*, respectively.
Figure 3.11 shows the created mesh. The tuyere is located in the bottom right corner and has a length of 50 *mm*. Figure 3.10 shows a close up of the tuyere region. In the area around the tuyere, this part of the mesh has the finest structure. The cell length increases from 1.2 *mm* to a maximum of 30 *mm* to reduce the cell number in areas were higher resolution is not necessary. The total cell number is 2498.

In the first step of the simulation, the particles were filled into the geometry from the top. For easier filling, the geometry was filled higher than the highest particle level of 850 *mm* and the surplus particles were removed afterwards.

After the filling process had been completed, the second step was the actual raceway simulation. This was performed by blowing air through the tuyere at a superficial velocity of 0.6 m/s. To find the best parameters for the simulation, eight test simulations were set up according to an experimental matrix. All possible combinations of the parameters `packingmodel` (implicit or explicit), the `momentumPredictor` turned on or off, and whether or not under-relaxation of air velocity, pressure and fluid volume fraction is active, were tested. Table 3.9 displays the test matrix.
Experiment 4 showed the best performance according to stability and time and therefore the settings `packingmodel: explicit`, `momentumPredictor: on` and no `underrelaxation` were used for the actual simulations.

Table 3.9: Experimental matrix for the raceway simulations. X meaning "activated", O meaning "not activated".

name	**implicit**	**explicit**	**momentum-Predictor**	**under-relaxation**
Experiment 1	X	O	X	X
Experiment 2	O	X	O	X
Experiment 3	X	O	X	O
Experiment 4	O	X	X	O
Experiment 5	X	O	O	X
Experiment 6	O	X	X	X
Experiment 7	X	O	O	O
Experiment 8	O	X	O	O

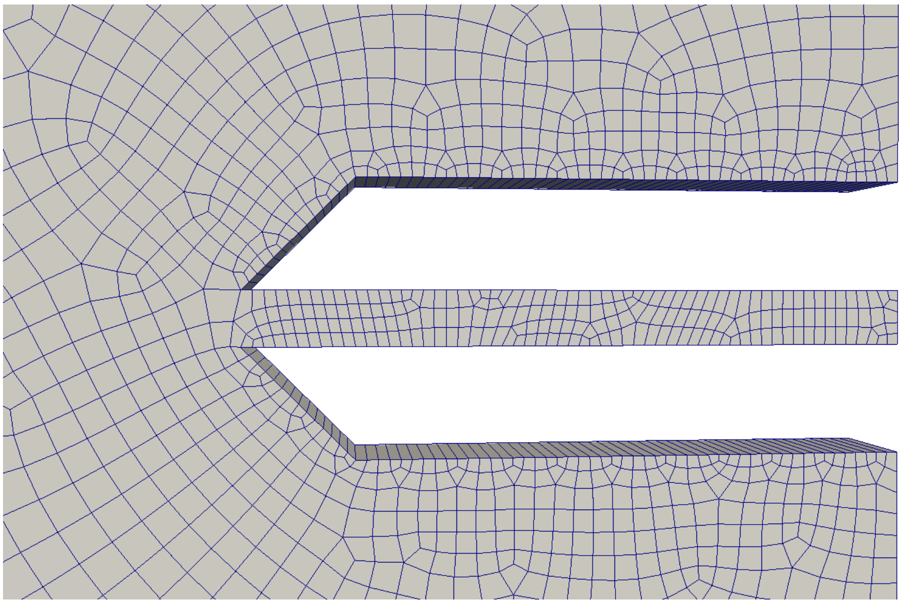


Figure 3.10: Detailed image of the tuyere region of the mesh for the first raceway validation case.

3.5.2.2 Raceway Simulation 2

Because the first raceway simulation gave no satisfying results (see subsection 4.2.1), a second evaluation was done by reproducing the work of [Feng et al., 2003].
They simulated gas-solid flow in a packed bed in a similar way as described in the previous section, but with following major differences:

- The created rectangular mesh has no tuyere area
- All mesh cells are of the same size.

The purpose of this simulation is to produce a raceway formation with the MPPIC-Foam solver for a simple geometry without testing different drag models. Table 3.10 displays the simulation parameters:

Table 3.10: Simulation parameters for the second raceway evaluation.

Parameter	**Unit**	**Value**
Number of particles	-	46000
Density	kg/m^3	2500
Diameter	m	0.004
Bed height	m	1
Bed width	m	0.3
Bed thickness	m	0.0162
Jet opening	m	0.02
Jet center above bottom	m	0.1
Cell width	m	0.005
Cell height	m	0.005
Cell volume	m^3	$4.05 \cdot 10^{-7}$

Early simulations showed, that the mesh cells are still too small for the MPPICFoam solver to handle, since the particle diameter is only 1 mm smaller than the length of a mesh cell. Therefore, the height and width of the cells were enlarged to 10 mm each. Figure 3.12 shows the mesh.
Instead of using a tuyere, the air flow is provided at a patch in the side wall 0.01 m above the bottom of the geometry. The air is injected with a velocity of 20 m/s for a duration of 3.5 s. The `packingmodel` was set to `explicit`. No working setup for the type `implicit` could be found. No `dampingmodel` was used and the simulation was carried out with the Wen-Yu drag model. No equations were under-predicted.

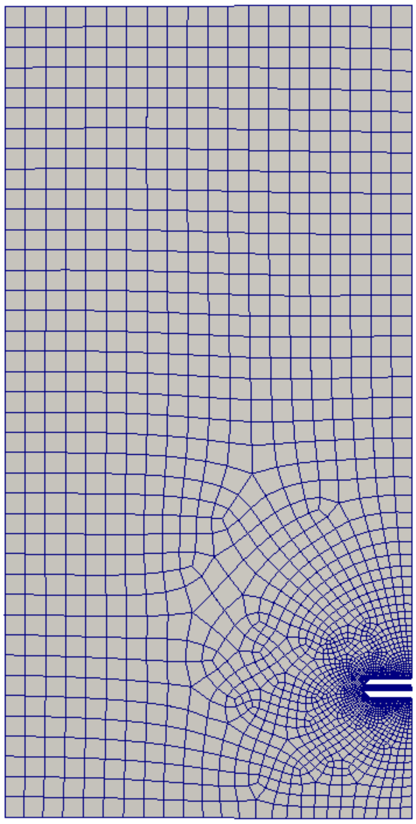


Figure 3.11: Mesh of the first raceway validation case with the tuyere in the lower right corner.

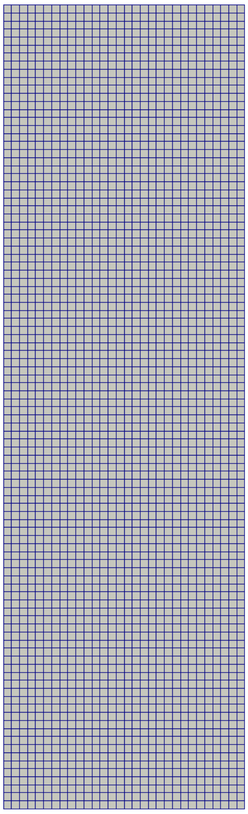


Figure 3.12: Mesh of the second raceway validation case.

4 Results and Discussion of MPPICFoam Simulations

4.1 Fluidized Bed Case

Figure 4.1 compares the results of the OpenFOAM simulations to the experimental data from [Chladek et al., 2017].

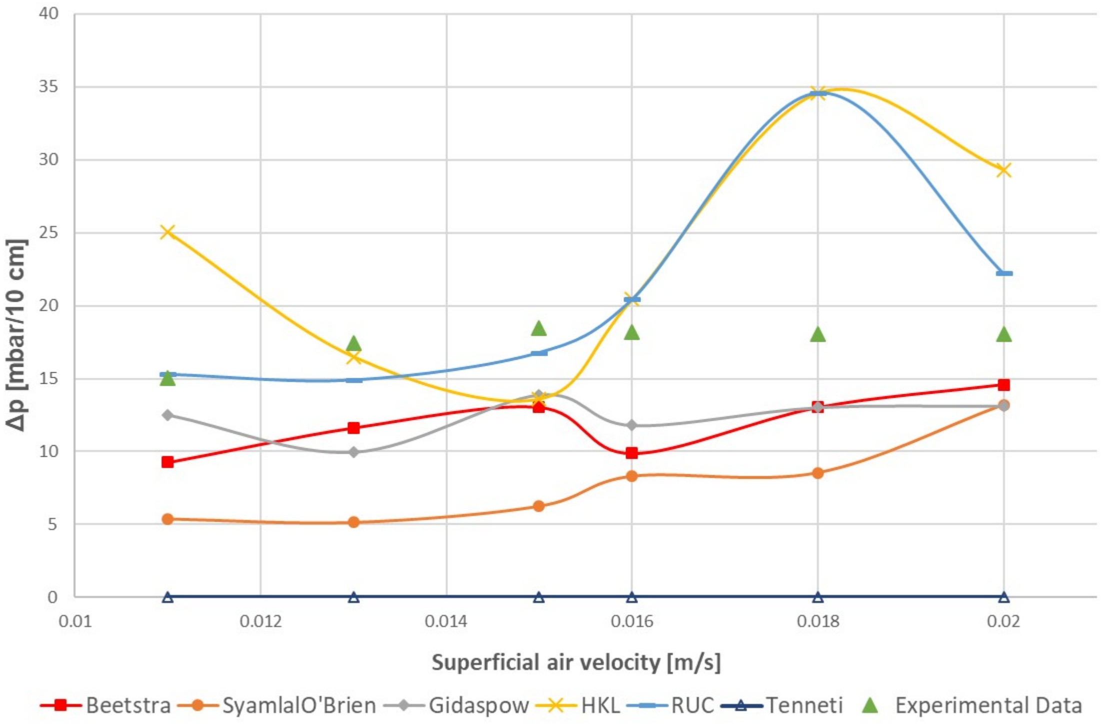


Figure 4.1: Change of the pressure drop in a fluidized bed with increasing superficial air flow rate. Experimental data by [Chladek et al., 2017].

In general, no drag model fits the experimental results over the whole range of tested superficial air velocities. Unfortunately, the Tenneti drag model does not predict any pressure loss at all. It is possible that there is an error in the implementation. Even after several checks of the simulation setup, the reason for this behavior was not found.

For the first three simulations, with superficial air velocities below the minimum fluidization velocity of 0.016 m/s, the fluidized bed is in the state of a fixed bed. In this range, the RUC drag model shows the best agreement with the experiment. The

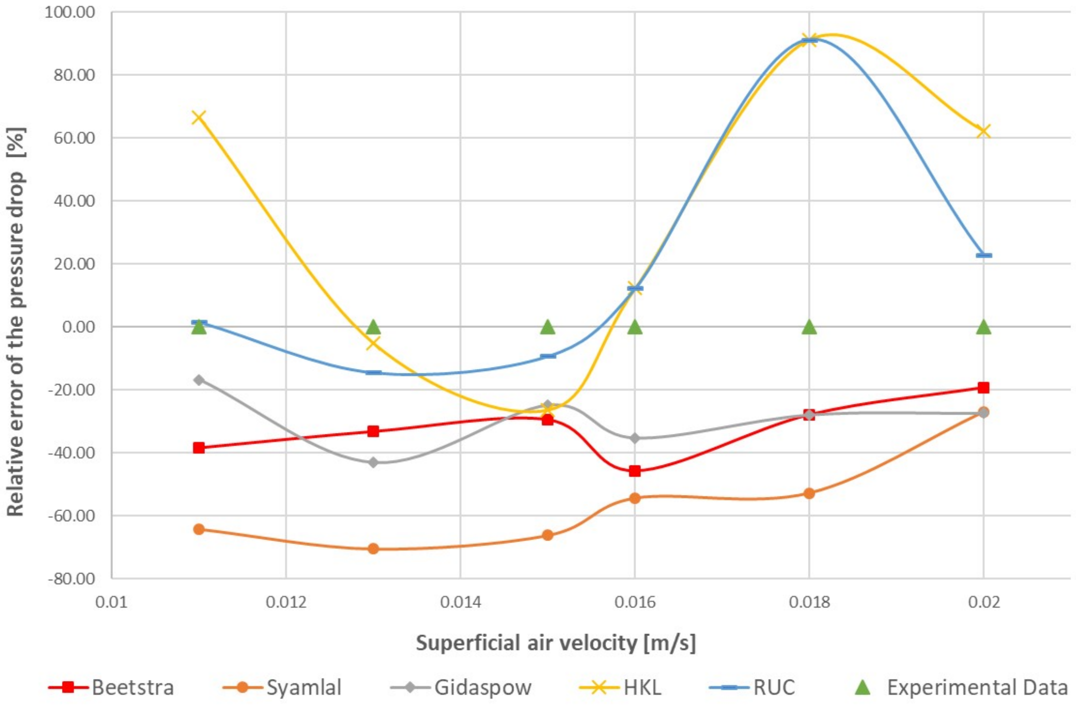


Figure 4.2: Relative deviation of the fluidized bed simulation's pressure drop (Chladek = 0). Experimental data by [Chladek et al., 2017].

Beetstra and Gidaspow drag model underpredict the pressure drop in this section. Considering the fact that the RUC drag model is basically the Ergun drag model with different coefficients, and that the Gidaspow model uses the Ergun drag model in this range of solid volume fractions, the results are reasonable. The HKL drag model shows good agreement with the experimental data for the simulations with 0.013 m/s and 0.015 m/s, but overpredicts the pressure drop for 0.011 m/s. The Syamlal O'Brien drag model predicts the lowest values. The reason for this behavior could be that the Syamlal O'Brien drag model is derived for a single spherical particle in a fluid and does not account for swarm effects.

When the velocity reaches 0.016 m/s the system changes from a fixed bed to a bubbling bed, which is the most difficult section to predict for drag models. Therefore, the predictions of the drag model vary the most for the simulations with superficial air velocities of 0.018 m/s and 0.02 m/s. The HKL and the RUC model predict higher pressure losses than the experimental data. The simulations using the drag models of Gidaspow and Beetstra show lower results. In general, these two drag models calculate the most invariant pressure drop. The Syamlal O'Brien drag model predicts values closer to the experimental data for this flow regime and seems to approach the experimental data at increasing superficial air velocity. The improvement of the Syamlal O'Brien model could be a result of the decrease in solid volume fractions at higher velocities.

For better comparison, Figure 4.2 shows the simulation results as relative deviations from the experimental data. It shows deviations of up to 70% for the 0.013 m/s test case, which increased up to 90% for a velocity of 0.018 m/s. For the 0.015 m/s test case, the models agree best with a deviation of 30% except for the Syamlal O'Brien drag model, which has not been derived for a fixed bed [Syamlal and O'Brien, 1987].

The results section of [Jayarathna et al., 2017] shows a comparison of the filling heights of the fluidized beds at different superficial air velocities. That was also planned for this evaluation, but no differences between the heights of the beds are visible and therefore the figure is not displayed here.

4.2 Raceway Case

4.2.1 Raceway Simulation 1

The results of the first raceway test cases defined in subsection 3.5.2 are shown in Figure 4.5 to Figure 4.6. All figures belong to the simulation with a particle filling height of 450 mm. Shortly after the start of the simulation a major problem occurred. As seen in Figure 4.5, an unnatural cavity developed around the tuyere. The reason for this behavior is the small cell volume in this area of about $6 \cdot 10^{-9}\ m^3$, whereas the volume of one parcel is about $2.7 \cdot 10^{-7}\ m^3$. The small cell volume is necessary to capture all fluid behavior reasonably in this area. As described in subsubsection 2.2.3.2, the cell volume has to be significantly larger than a computational parcel for mapping the parcel to a point on the grid.
Even at the highest superficial velocity of 0.6 m/s, the particles are too far away for the air blast to be affected. As a result, no proper raceway zone developed.
Figure 4.3 shows a detailed image of the tuyere area. It can clearly be seen, that only a few particles enter the zone around the tuyere, where the raceway should be.

Figure 4.4 shows the air velocity, the pressure and the fluid volume fraction over a horizontal line through the tuyere. The sampling line can be seen in Figure 4.5. The velocity decreases and increases rapidly inside the tuyere and decreases again after it passes the tip of the tuyere (at the first vertical line in Figure 4.4). Inside the cavity around the tuyere, the velocity stays constant. When the air reaches the particles (at the second vertical line in Figure 4.4), the velocity reduces and fluctuates until the end of the geometry is reached. It can be seen, that the fluid volume fraction is actually about 1 inside the cavity between the two black lines, meaning nearly all of the volume is air and no particles are present. Inside the particle filling, the value for α_f is slightly above 0.4. The pressure stays pretty constant for the whole cross section.

Figure 4.8 gives a visual representation of the air velocity in the tuyere region for six subsequent time steps. The red stream in 4.8a displays the air reaching the cavity. Figure 4.9 shows the pressure gradient for the same time steps in the simulation. Both figures show odds for the corresponding values in these time steps. Especially the visualization of the air flow shows quite unnatural behavior. The first picture shows the steady air flow with a stream coming out of the tuyere like expected, but in the next figure the velocity is much less and in the third, the air flow is nearly gone. The fourth figure shows results similar to the first one, but with lower velocity, while in the last two pictures, the air flow creates layers inside the tuyere. The pressure for the same time steps fluctuates as well. The particle filling does not seem to have any effect on the pressure at all. Once again, the small cells in the tuyere region seem to be the reason for this behavior.

In Figure 4.6, the fluid volume fraction for the whole geometry is shown. Once again, the red zones around the tuyere in the bottom right corner indicate the lack of particles in this area.
Figure 4.7 shows the trajectories of the particles for a time span of 2 s. It shows, that the particles in the lower part of the filling (red and yellow pathlines) rise, while the ones in the top layers (blue pathlines) move towards the tuyere. In the tuyere region, some particles move wildly around and even seem to go through the tuyere, which occurs because the trajectories are made only from the written time steps.

Overall no reasonable were obtained for this case. The MPPICFoam solver predicts unnatural pressure and flow (see 4.8a and Figure 4.9).

4.2.2 Raceway Simulation 2

The results of the second raceway simulations are shown in Figure 4.10 as a time series. In this case, the MPPICFoam solver managed to predict a raceway formation. In 4.10a, one bubble of air is already rising, while another one is formed at the jet entrance. The following figures (a-d) show how the air bubble rises and increases in size. In 4.10d it reaches the surface, lifting the surface layer of particles. In 4.10f, the surface is leveled again and the next air bubble is approaching the surface. The blue color shows, that the fluid volume fraction in the particle filling is about 0.4, which is a reasonable value.
Figure 4.11 displays the trajectories of the particles for the second raceway case. It shows that in the area above the jet entrance the particles move a lot in both upwards and downwards direction, while in the bottom and on the opposite side of the entrance the particles hardly move at all. The particle movement above the jet entrance occurs due to the wall effect, where the air bubbles rise next to the wall.
Figure 4.12 shows a comparison of the MPPICFoam results and the ones from [Feng et al., 2003] for an air velocity of 20 m/s. Predicted and experimental results show reasonable agreement.

Although this simulation gives quite satisfying results, still some issues occurred. After the simulation start, the particle filling is very unstable with "explosions" at the surface. "Explosions" are defined as unnatural sudden ejections of particles from the bed surfaces. At $t = 1.5$ s, the particles calm down and the simulation runs stable for 3.5 s. After that, the unstable behavior starts again until the end of the calculation. In the first three snapshots of Figure 4.10, some particles coming back down from such an "explosion" are visible right above the particle filling (slightly yellow areas).

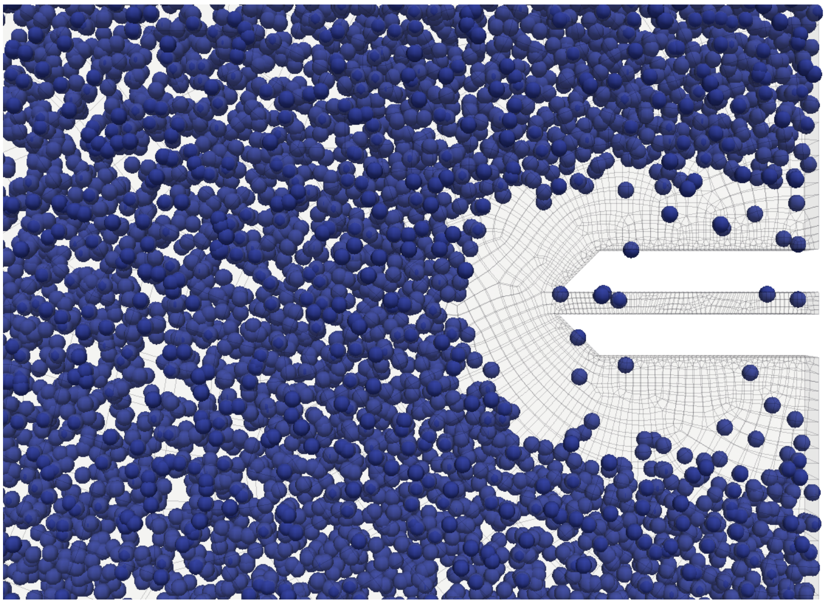


Figure 4.3: Detailed image of the tuyere area of the raceway case domain.

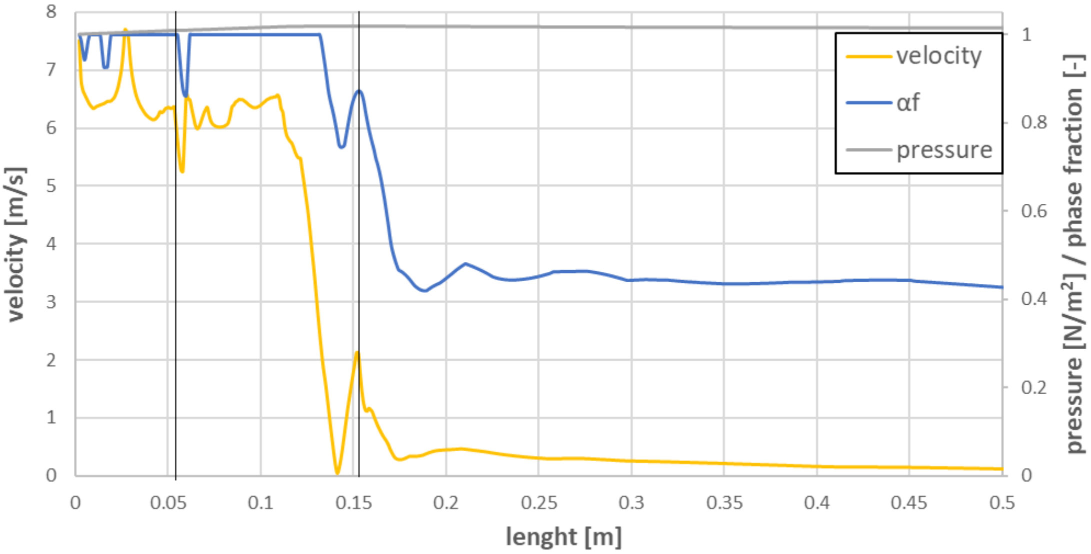


Figure 4.4: α_f, air velocity and pressure over a straight horizontal line trough the geometry at the tuyere height (see Figure 4.5). First black line: tuyere tip, second black line: particle bed. The origin is at the inlet.

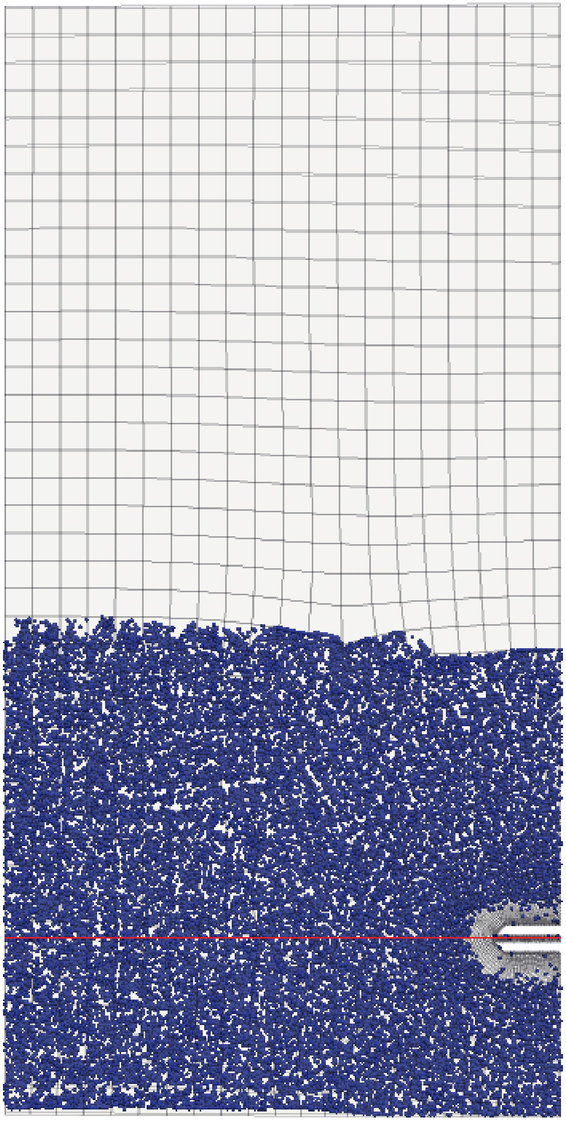


Figure 4.5: Visualization of the filled mesh for the raceway case. The plot over the red line can be seen in Figure 4.4.

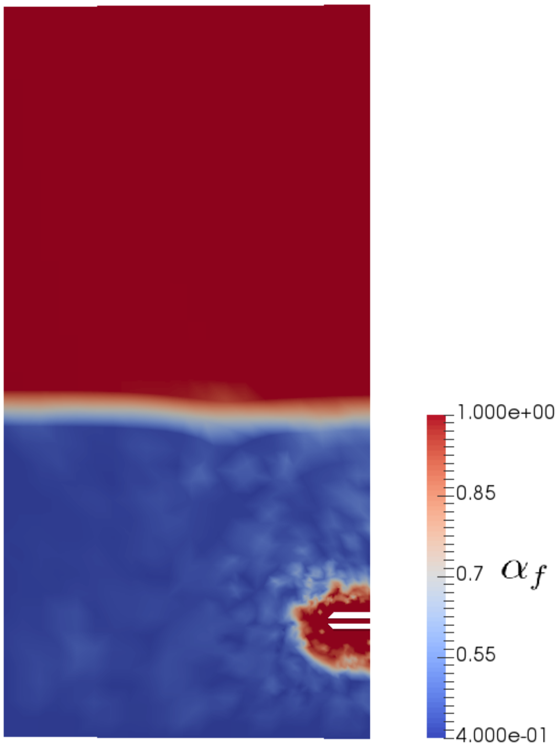


Figure 4.6: Visualization of α_f for the first raceway case.

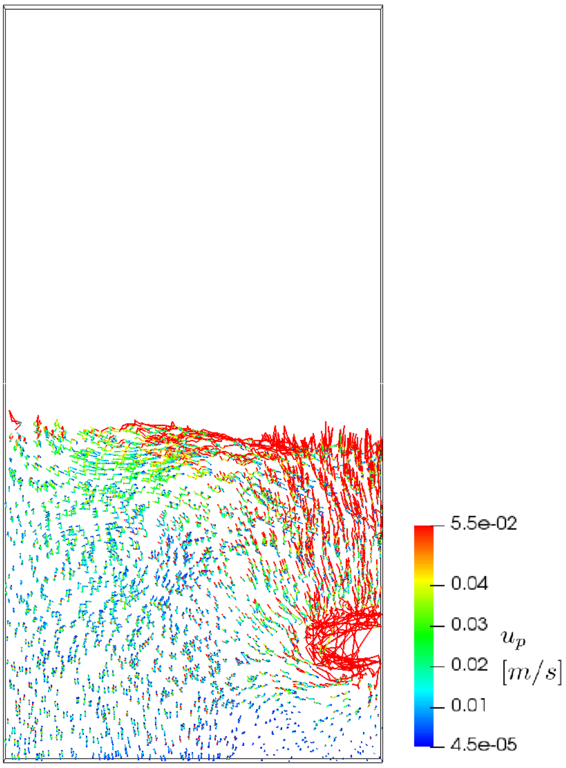
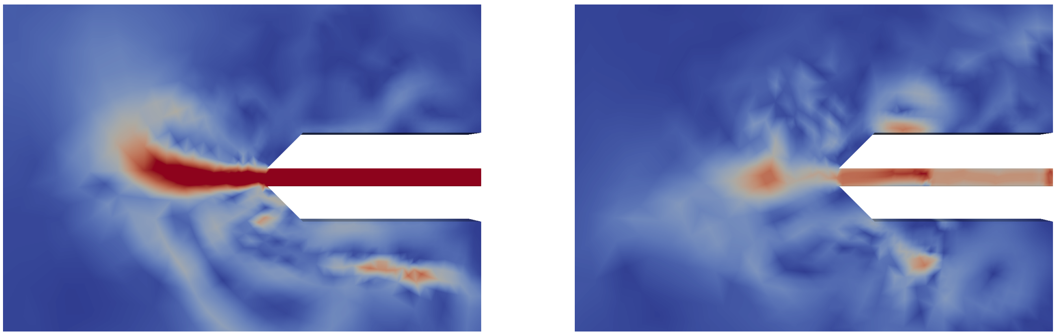
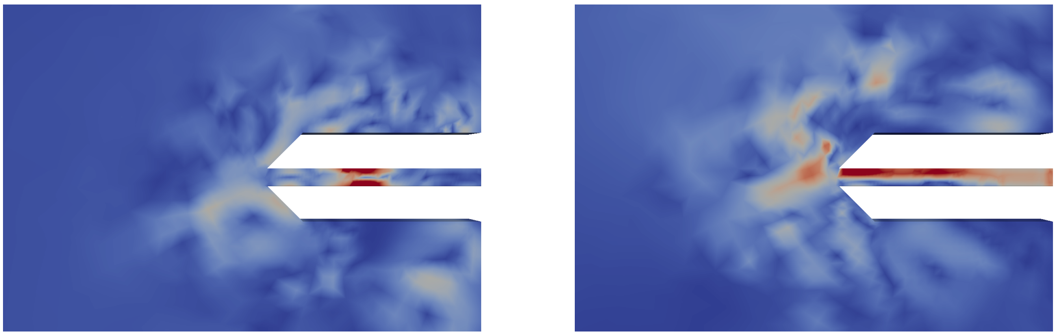


Figure 4.7: Visualization of the trajectories of the particles for the first raceway case.

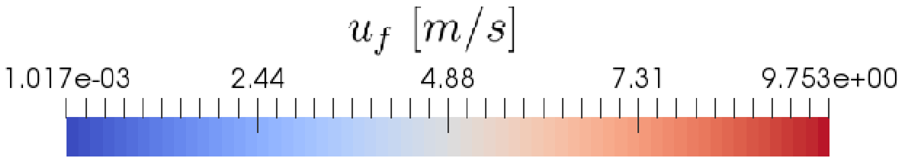


(a) t = 0.1 s (b) t = 0.2 s

(c) t = 0.3 s (d) t = 0.4 s

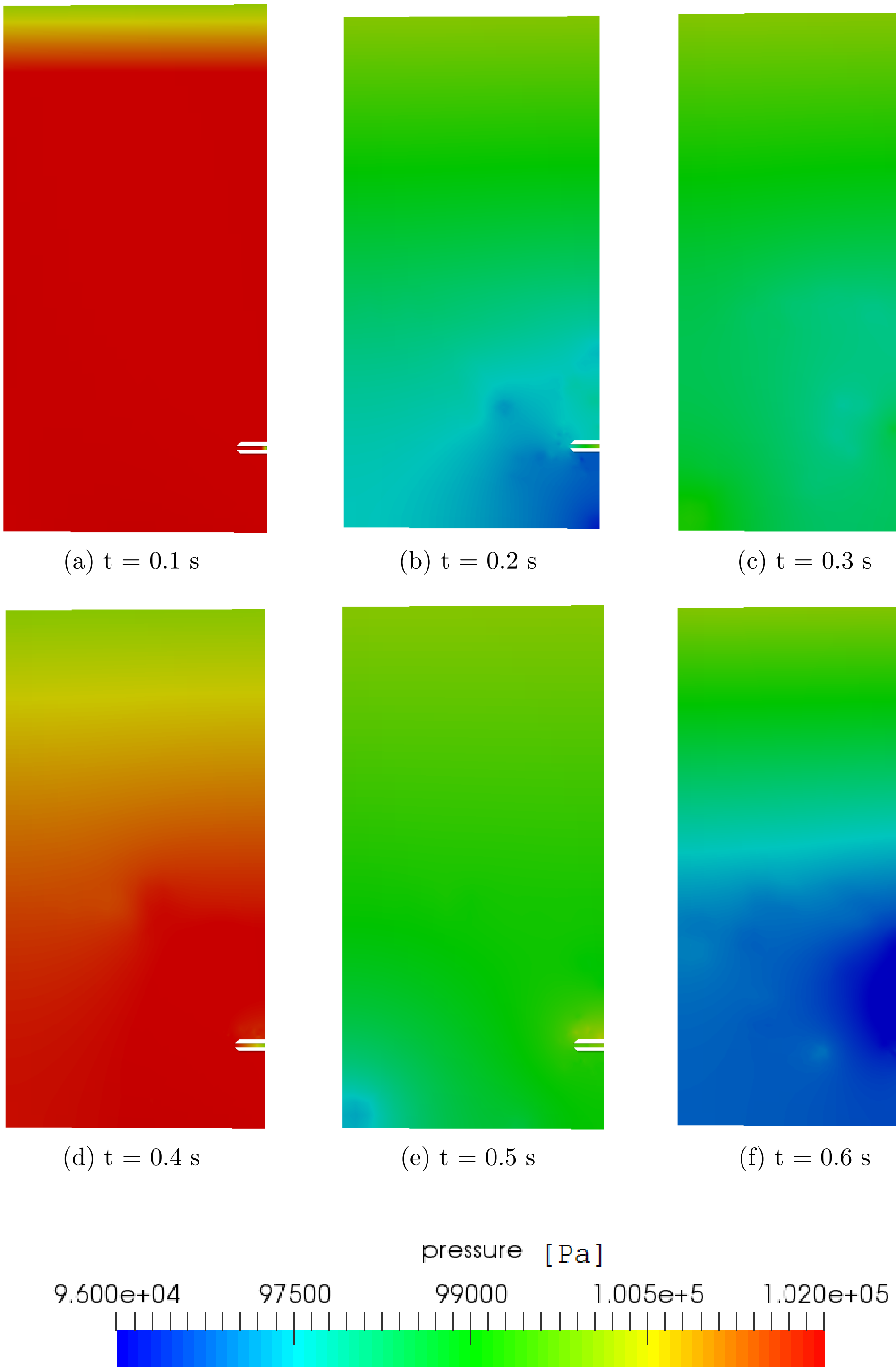


(e) t = 0.5 s (f) t = 0.6 s



(g) legend

Figure 4.8: Snapshots of u_f for six time steps of the first raceway simulation. Deficient results due to errors in the MPPIC method caused by small mesh cells.

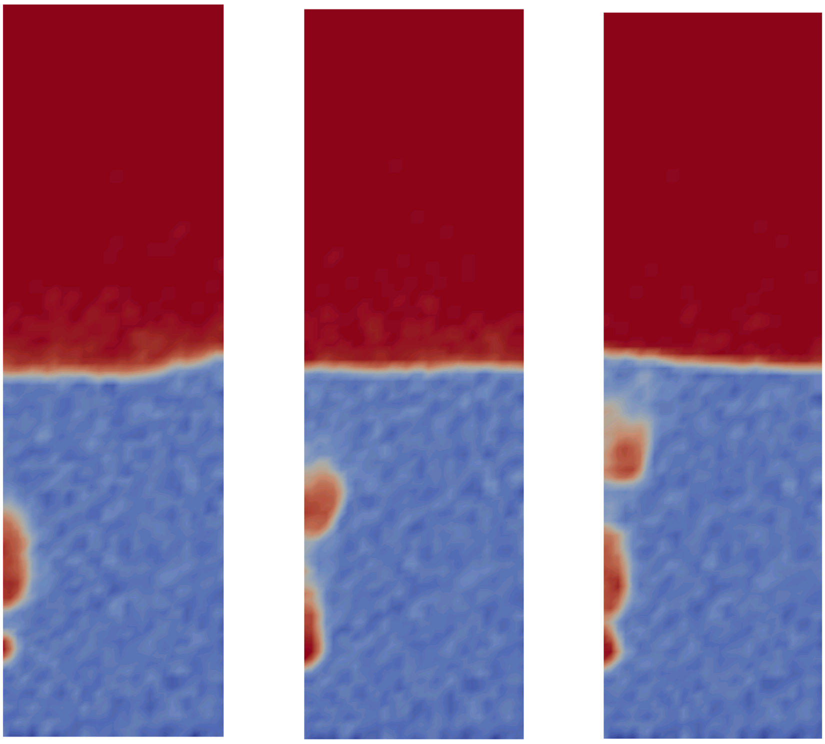


(a) t = 0.1 s (b) t = 0.2 s (c) t = 0.3 s

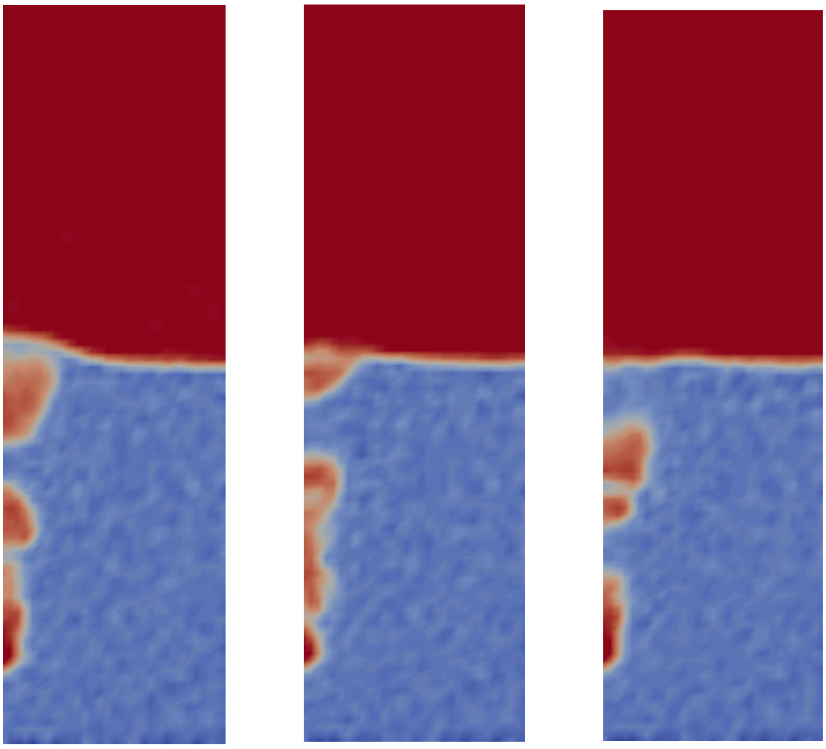
(d) t = 0.4 s (e) t = 0.5 s (f) t = 0.6 s

(g) legend

Figure 4.9: Snapshots of p for six subsequent time steps of the first raceway simulation. Deficient results due to errors in the MPPIC method caused by small mesh cells.



(a) t = 1.5 s (b) t = 1.6 s (c) t = 1.7 s



(d) t = 1.8 s (e) t = 1.9 s (f) t = 2 s

(g) legend

Figure 4.10: Snapshots of α_f for six subsequent time steps of the second raceway simulation.

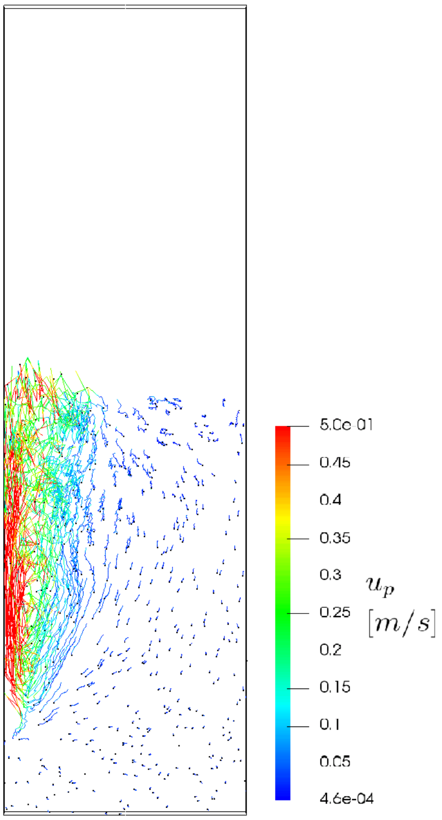


Figure 4.11: Visualization of the trajectories of the particles for the second raceway case.

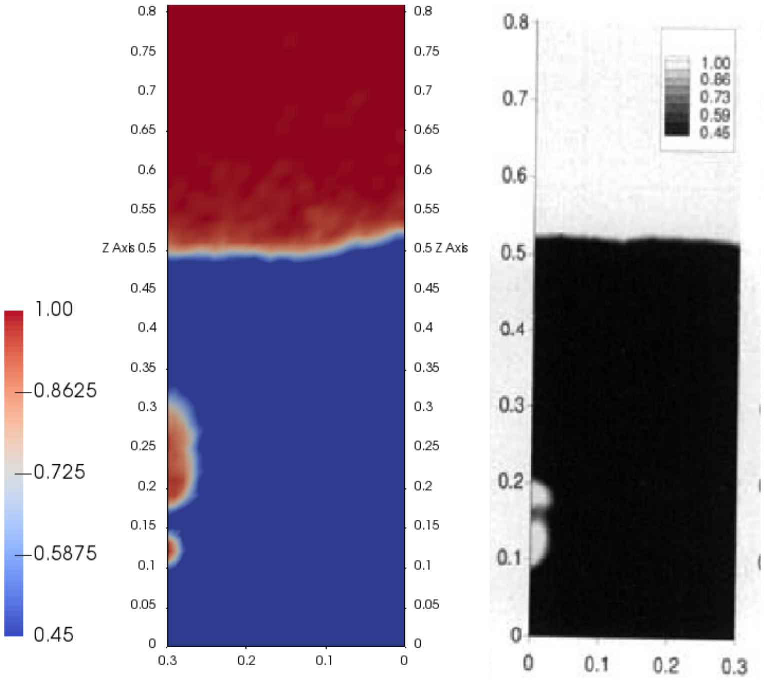


Figure 4.12: Comparison of predicted and simulated α_f. Left: MPPICFoam simulation, right: Results from [Feng et al., 2003].

5 Conclusion

A study on Euler-Lagrange (EL) hybrid models handling slowly moving and fixed beds using different drag models has been carried out. There has been a comparison and additional evaluation of different drag models. A filling of a randomly packed bed with the DPM hybrid approach [Adamczyk et al., 2014] and a series of evaluations of slowly moving beds and the creation of a raceway zone in a particle filling with the hybrid MPPIC approach [Andrews and O'Rourke, 1996] were investigated.

Simple test cases show, that for raceway conditions, the RUC drag model [Du Plessis and Masliyah, 1988], the drag models of Beetstra [Beetstra et al., 2007], HKL [Hill et al., 2001] [Benyahia et al., 2006], and Gidaspow [Gidaspow, 1994] give realistic predictions for the momentum exchange coefficient. The predictions according to the Tenneti [Tenneti et al., 2011] and Richardson Zaki [Richardson and Zaki, 1997] models deviate significantly from the others.

The drag models were additionally tested by comparing predicted pressure drops with the experimental data from [Beetstra et al., 2007]. All used drag models predict reasonable pressure drops for varying sphere sizes. Since the Richardson Zaki model failed in the first comparison, it was not used for this evaluation. The models by Tenneti, Gidaspow, and the RUC model give the best results for the experiments of [Beetstra et al., 2007]. Generally, the deviation of the calculated values increases at larger particle diameters.

The ability of the DPMFoam solver of handling packed beds was tested by creating a randomly packed bed and comparing the results with DEM simulations and experimental data by [Haddadi et al., 2017]. The evaluation showed, that DPMFoam simulations agree well with the results of [Haddadi et al., 2017]. The simulated beds are packed in a similar way and the difference of the particle number between the experiment and the DPMFoam simulation is less than 2%. Reasons for the lower value of the DPMFoam calculations may be the not perfectly circular cylinder, errors in the particle-particle interactions (space between two particles is sometimes too wide) and mistakes in the manual counting process.

EL hybrid models in combination with the chosen drag models were evaluated with fluidized bed simulations by [Jayarathna et al., 2017]. For these calculations, less promising results were made. In general, no drag model used with MPPICFoam fits the experimental results over the tested range of superficial air velocities. The RUC model gives best results below the fluidization point, while the models by Gidaspow and Beetstra agree best in the bubbling bed regime. Maybe due to an error in the implementation, the Tenneti model predicts zero pressure loss. MPPICFoam seems to have troubles for very dense gas-particle flows, because particle-particle interactions become unstable and lead to unnatural bed explosion.

The last step was the evaluation of the capabilities to predict raceway formation using experimental results from [Mojamdar et al., 2018] and [Feng et al., 2003]. A major problem concerning the computational mesh occurred with the raceway simulations in the Mojamdar case. Due to the small cell size in the tuyere region, hardly any particles enter this area. However, the small cell volume is required to capture the fluid flow correctly in this area. Thus, no viable results were made for these experiments. The cavity around the tuyere was significantly larger than in the experiments. The mesh seems to influence the raceway formation. A second evaluation with a simpler mesh and cells which are significantly larger than the particles was done using the work of [Feng et al., 2003]. For this case, the MPPICFoam solver gave acceptable results. This shows that the MPPICFoam solver has issues with small mesh cells.

There are limitations of the current OpenFOAM implementation of the MP-PIC method. For instance, particles are assumed to be perfect spheres. Corrections for non-spherical particles can only be made via the drag correlation. Furthermore, the grid cell size has to be significant larger than the particle size for accurate interpolation.

6 Outlook

Given that the MPPICFoam simulations did not give satisfying results, future experiments should be carried out with conditions that are more like blast furnaces. For these simulations, cell volumes should be notably bigger than the particle volumes. Alternatively, a different way to cope with parcels bigger than cells has to be found.
The DPMFoam simulation showed, that the creation of a random packed bed could be carried out in the future with this solver to save computational time.

Furthermore, the Euler-Lagrange hybrid models should be extended to handle reacting particles for realistic raceway zone simulations.

Bibliography

[Adamczyk et al., 2014] Adamczyk, W. P., Klimanek, A., Białecki, R. A., Węcel, G., Kozołub, P., and Czakiert, T. (2014). Comparison of the standard Euler-Euler and hybrid Euler-Lagrange approaches for modeling particle transport in a pilot-scale circulating fluidized bed. Particuology, 15:129–137.

[Akgiray and Saatci, 1998] Akgiray, O. and Saatci, A. (1998). A Critical Look at Declining Rate Filtration Design. Water Science and technology, (38):89–96.

[Andrews and O'Rourke, 1996] Andrews, M. and O'Rourke, P. (1996). The multiphase particle-in-cell (MP-PIC) method for dense particulate flows. International Journal of Multiphase Flow, 22(2):379–402.

[Batchelor, 2000] Batchelor, G. K. (2000). An Introduction to Fluid Dynamics. Cambridge University Press.

[Bauer et al., 2016] Bauer, M., Morton, J., Munro, C., and Autterson, B. (2016). ANSYS Fluent 17.0 User Guide.

[Beetstra et al., 2007] Beetstra, R., van der Hoef, M., and Kuipers, H. (2007). Drag Force of Intermediate Reynolds Number Flow Past Mono- and Bidisperse Arrays of Spheres. AIChE Journal, 53:489–501.

[Benyahia et al., 2006] Benyahia, S., Syamlal, M., and O'Brien, T. J. (2006). Extension of Hill-Koch-Ladd drag correlation over all ranges of Reynolds number and solids volume fraction. Powder Technology, 162(2):166–174.

[Brennen, 2005] Brennen, C. E. (2005). Fundamentals of Multiphase Flow. Cambridge University Press.

[Bunge, 2003] Bunge, U. (2003). Der SIMPLE-Algorithmus. TU Berlin, Berlin.

[Burgess, 1985] Burgess, J. (1985). Fuel Combustion in the Blast Furnace Raceway Zone. Prog. Energy Combust. Sci., 11:61–82.

[Burns et al., 2004] Burns, A. D., Frank, T., Hamill, I., and Shi, J.-M. (2004). The Favre Averaged Drag Model for Turbulent Dispersion in Eulerian Multi-Phase Flows. In 5th International Conference on Multiphase Flow, ICMF 04 Yokohama, Japan, May 30 June 4, 2004.

[Chladek et al., 2017] Chladek, J., Jayarathna, C. K., Tokheim, L. A., and Moldestad, B. M. E. (2017). Classification of Solid Particles with Size and Density Difference in a Gas Fluidized Bed. Chemical Engineering Science, (177):151–162.

[Crowe et al., 1998] Crowe, C. T., Sommerfeld, M., and Tsuji, Y. (1998). Multiphase flows with droplets and particles. CRC Press.

[Dalla Valle, 1943] Dalla Valle, J. M. (1943). Micromeritics, the Technology of Fine Particles. Pitman.

[Dittmann, 1995] Dittmann, A. (1995). Repetitorium der Technischen Thermodynamik. Teubner B.G. GmbH.

[Du Plessis and Masliyah, 1988] Du Plessis, J. P. and Masliyah, J. H. (1988). Mathematical modelling of flow through consolidated isotropic porous media. Transport in Porous Media, 3(2):145–161.

[Elghobashi, 1994] Elghobashi, S. (1994). On predicting particle-laden turbulent flows. Applied scientific research, 52:309–329.

[Ergun, 1952] Ergun, S. (1952). Fluid flow through packed columns. Carnegie Institute of Technology, 48(2):89 f.

[Feng et al., 2003] Feng, Y.-q., Pinson, D., Yu, A.-b., and Zulli, P. (2003). Numerical Study of Gas-Solid Flow in the Raceway of a Blast Furnace. Steel Research, 74:523–530.

[Garside and Al-Dibouni, 1977] Garside, J. and Al-Dibouni, M. (1977). Velocity-Voidage Relationships for Fluidization in Solid-Liquid Systems. Industrial & Engineering Chemistry Process Design and Development, 16(2):206–214.

[Geldart, 1973] Geldart, D. (1973). Types of Gas Fluidization. 7:285–292.

[Gidaspow, 1994] Gidaspow, D. (1994). Multiphase Flow and Fluidization: Continuum and Kinetic Theory Descriptions. Academic Press.

[Golas, 1999] Golas, P. J. (1999). Science and Civilisation in China: Volume 5, Chemistry and Chemical Technology, Part 13, Mining. Cambridge University Press, page 152.

[Haddadi et al., 2017] Haddadi, B., Jordan, C., Norouzi, H. R., and Harasek, M. (2017). Numerical Investigation of Particle Types Influence on Packed bed Adsorber Behaviour. Technical report, Technische Universität Wien.

[Harris and Crighton, 1994] Harris, S. and Crighton, D. (1994). Solitons, solitary waves, and voidage disturbances in gas-fluidized beds. Journal of Fluid Dynamics, 266:243–276.

[Hill et al., 2001] Hill, R. J., Koch, D. L., and Ladd, A. J. C. (2001). The first effects of fluid inertia on flows in ordered and random arrays of spheres. Journal of Fluid Mechanics, 448:213–241.

[Honerkamp et al., 2012] Honerkamp, J., Hartmann, and Römer (2012). Klassische Theoretische Physik. Springer-Verlag Berlin Heidelberg.

[Igci et al., 2008] Igci, Y., Andrews, A. T., Sundaresan, S., and Brien, T. O. (2008). Filtered Two-Fluid Models for Fluidized Gas-Particle Suspensions. AlChE Journal, 54(6):1431–1448.

[Jakobsen, 2008] Jakobsen, H. A. (2008). Chemical Reactor Modeling - Multiphase reactive Flows. Springer.

[Jayarathna et al., 2017] Jayarathna, C., Moldestad, B., and Tokheim, L.-A. (2017). Validation of results from barracuda cfd modelling to predict the minimum fluidization velocity and the pressure drop of geldart a particles. Proceedings of the 58th SIMS, pages 76–82.

[Kuhlmann, 2007] Kuhlmann, H. (2007). Strömungsmechanik. Pearson Studium.

[Lerner and Trigg, 1991] Lerner, R. G. and Trigg, L. (1991). Enciclopaedia of Physics (2nd edition). VHC publishers.

[Lun et al., 1984] Lun, C. K. K., Savage, S., Jeffrey, D., and Chepurniy, N. (1984). Kinetic theories for granular flow: Inelastic particles in couette flow and slightly inelastic particles in a general flowfield. Journal of Fluid Mechanics, 140:223–256.

[Lundberg J., 2008] Lundberg J., H. B. M. (2008). A Review of some existing Drag Models describing the Interaction between Phases in a Bubbling Fluidized Bed. Master's Thesis, Telemark University College.

[McCormick, 1979] McCormick, B. W. (1979). Aerodynamics, Aeronautics, and Flight Mechanics. John Wiley & Sons, Inc.

[McGraw-Hill, 1994] McGraw-Hill (1994). Encyclopedia of Physics. C. B. Parker.

[Mojamdar et al., 2018] Mojamdar, V., Gupta, G. S., and Puthukkudi, A. (2018). Raceway Formation in a Moving Bed. 58(8):1396–1401.

[Moukalled et al., 2015] Moukalled, F., Mangani, L., and Darwish, M. (2015). The finite volume method in computational fluid dynamics - an advanced introduction with openfoam and matlab. Springer.

[Norouzi et al., 2016] Norouzi, H. R., Zarghami, R., Sotudeh-Gharebargh, R., and Mostoufi, N. (2016). Coupled CFD-DEM Modeling: Formulation, Implementation and Application to Multiphase Flows. John Wiley & Sons, Ltd.

[O'Rourke and Snider, 2010] O'Rourke, P. J. and Snider, D. M. (2010). An improved collision damping time for MP-PIC calculations of dense particle flows with applications to polydisperse sedimenting beds and colliding particle jets. Chemical Engineering Science, 65(22):6014–6028.

[O'Rourke and Snider, 2012] O'Rourke, P. J. and Snider, D. M. (2012). Inclusion of collisional return-to-isotropy in the MP-PIC method. Chemical Engineering Science, 80:39–54.

[O'Rourke et al., 2009] O'Rourke, P. J., Zhao, P. P., and Snider, D. (2009). A model for collisional exchange in gas/liquid/solid fluidized beds. Chemical Engineering Science, 64(8):1784–1797.

[Richardson and Zaki, 1997] Richardson, J. F. and Zaki, W. N. (1997). Sedimentation and fluidisation: Part I. Chemical Engineering Research and Design, 75:S82 – S100.

[Rowe et al., 1960] Rowe, P., Henwood, G., and Authority, U. K. A. E. (1960). Drag Forces in a Hydraulic Model of a Fluidized bed, Part II. UKAEA.

[Shiller and Naumann, 1935] Shiller, L. and Naumann, A. (1935). A Drag Coefficiant Correlation. Zeitschrift des Vereins Deutscher Ingenieure, 77:318–320.

[Snider, 2001] Snider, D. M. (2001). An Incompressible Three-Dimensional Multiphase Particle-in-Cell Model for Dense Particle Flows. Journal of Computational Physics, 170(2):523–549.

[Syamlal and O'Brien, 1987] Syamlal, M. and O'Brien, T. (1987). The Deviation of a Drag Coefficient Formula from Velocity-Voidage Correlations. Unpublished report.

[Tenneti et al., 2011] Tenneti, S., Garg, R., and Subramaniam, S. (2011). Drag law for monodisperse gas-solid systems using particle-resolved direct numerical simulation of flow past fixed assemblies of spheres. International Journal of Multiphase Flow, 37(9):1072–1092.

[Travis and Harlow, 1976] Travis, J. and Harlow, F. (1976). Numerical calculation of two-phase flows. Nuclear Science and Engineering, 61:1–10.

[Versteeg and Malalasekera, 1995] Versteeg, H. and Malalasekera, W. (1995). An introduction to Computational Fluid Dynamics - The Finite Volume Method. Longman Scientific Technical.

[Weir and McGavin, 2008] Weir, G. and McGavin, P. (2008). The coefficient of restitution for the idealized impact of a spherical, nano-scale particle on a rigid plane. Proceedings of the Royal Society of London Series A, 464:1295–1307.

[Weller et al., 2018] Weller, H., Greenshields, C., and de Rouvray, C. (2018). OpenFOAM User Guide.

[Weller et al., 1998] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput. Phys., 12(6):620–631.

List of Figures

List of Tables