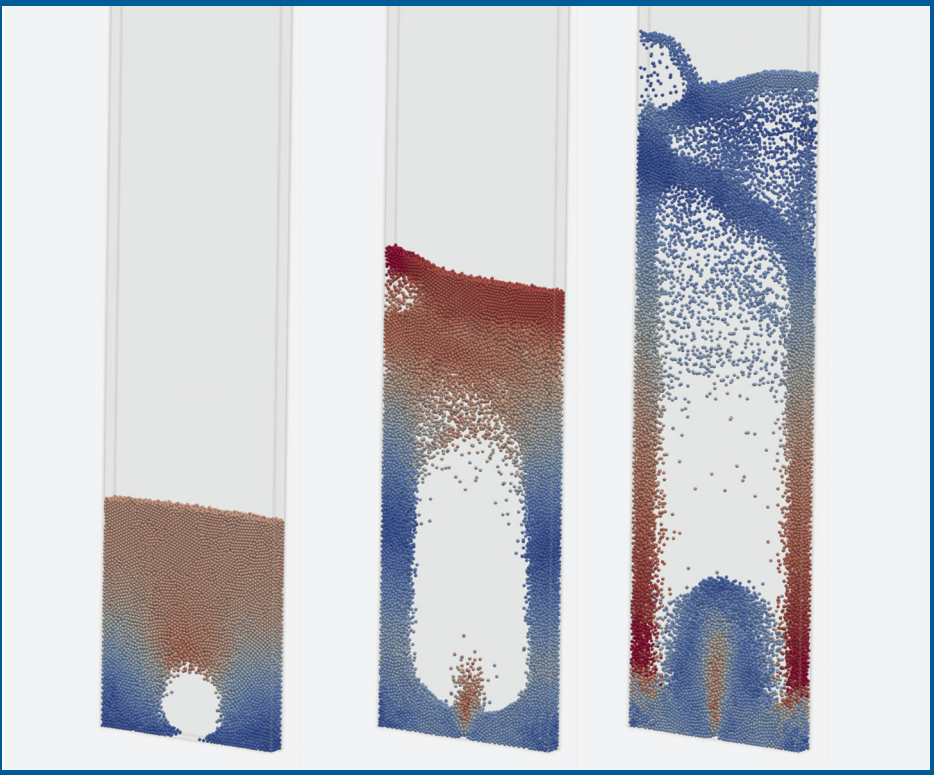
Jens Bender

Lagrangian-Lagrangian fluid-solid coupling in a generalized finite difference framework



Technische Universität Kaiserslautern

Fraunhofer ITWM

Fraunhofer Institute for
Industrial Mathematics ITWM

Lagrangian-Lagrangian fluid-solid coupling in a generalized finite difference framework

Jens Bender

FRAUNHOFER VERLAG

Contact:
Fraunhofer Institute for
Industrial Mathematics ITWM
Fraunhofer-Platz 1
67663 Kaiserslautern
Germany
Telefon +49 631/31600-0
E-Mail info@itwm.fraunhofer.de
URL www.itwm.fraunhofer.de

Bibliographic information of the German National Library:
The German National Library has listed this publication in its Deutsche Nationalbibliografie; detailed bibliographic data is available on the internet at www.dnb.de.

ISBN 978-3-8396-1674-1

D 386

Zugl.: Kaiserslautern, TU, Diss., 2020

Print and finishing:
Fraunhofer Verlag, Mediendienstleistungen

The book was printed with chlorine- and acid-free paper.

Nobelstrasse 12
70569 Stuttgart
Germany
verlag@fraunhofer.de
www.verlag.fraunhofer.de

is a constituent entity of the Fraunhofer-Gesellschaft, and as such has no separate legal status.

Fraunhofer-Gesellschaft zur Förderung
der angewandten Forschung e.V.
Hansastrasse 27 c
80686 München
Germany
www.fraunhofer.de

Lagrangian-Lagrangian fluid-solid coupling in a generalized finite difference framework

Jens Bender
Department of Mathematics,
University of Kaiserslautern, Germany.

Vom Fachbereich Mathematik
der Technischen Universität Kaiserslautern
genehmigte Dissertation
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)

Betreuer: Prof. Dr. Axel Klar, Technische Universität Kaiserslautern, Deutschland
Zweitgutachter: Prof. Dr. Jens Struckmeier, Universität Hamburg, Deutschland

Datum der Disputation: 29.06.2020

D386

Abstract

EN

We consider the modeling and simulation of flows composed of a fluid with an immersed particulate solid phase within a two-way coupled scheme. We embed it into the generalized finite difference framework of the finite pointset method (FPM). Both phases are described in a Lagrangian formalism and are represented by point clouds. This allows us to treat all phases in a common framework and to take advantage of synergies in terms of data structures and algorithms. A key challenge which is introduced by the generalized finite difference setting is the calculation of averaged quantities. Due to the properties of our mesh-free approach which is missing an inherent definition of cell volume, conventional averaging strategies from mesh-based schemes are not directly applicable. We employ an approach which circumvents these problems and takes the finite difference nature of the FPM into account. Additionally, we bring to light the required changes to a projection method for the fluid phase to incorporate the multi-phase setting. The solid phase solver, averaging scheme, and fluid solver are embedded into a coupled algorithm with a substepping procedure to improve efficiency.

DE

Wir beschreiben die Modellierung und Simulation von Strömungen bestehend aus einem Fluid und einer dispersen Feststoff-Phase innerhalb eines zweiseitig gekoppelten Verfahrens. Dieses wird in das verallgemeinerte Finite-Differenzen-Schema der Finite Pointset Methode (FPM) eingebettet. Beide Phasen werden dabei in einem Lagrangeschen Formalismus beschrieben und durch bewegte Punktewolken repräsentiert. Das erlaubt uns alle Phasen auf einer gemeinsamen Grundlage zu beschreiben und Synergien bei Datenstrukturen und Algorithmen auszunutzen. Dabei entstehen jedoch durch die Nutzung von verallgemeinerten Finiten Differenzen Herausforderungen in Bezug auf die Berechnung gemittelter Größen. Durch die Eigenschaften unserer gitterfreien Methode, in der eine natürliche Definition von Zell-Volumina fehlt, sind herkömmliche Mittelungsverfahren von gitterbasierten Methoden nicht direkt anwendbar. Dieser Umstand wird diskutiert und wir präsentieren ein Verfahren welches dieses Problem umgeht und sich für den Finite-Differenzen-Ansatz der FPM eignet. Weiterhin wird auf der Basis eines Projektionsansatzes ein numerisches Schema für die Fluid-Phase an die mehrphasige Problemstellung angepasst. Dabei werden die Verfahren für Feststoff-Phase, Mittelung und Fluid-Phase in einem gekoppelten Algorithmus mit Teilschritten in der Feststoff-Phase eingebettet.

Acknowledgments

First and foremost, I would like to thank the Fraunhofer ITWM for the financial support which made this work possible and for providing an excellent working environment.

I further want to thank my academic supervisor, Prof. Dr. Axel Klar, who always managed to put things into perspective and counteract the pessimism one so often falls victim to in phases of fruitless attempts.

I am equally grateful for all the discussions within my department at Fraunhofer ITWM. First and foremost, I want to thank Dr. Jörg Kuhnert, for his advice and all the time he took, while having a full desk himself. Special thanks also go to my colleague, and former fellow PhD student, Dr. Pratik Suchde, for countless discussions, his honest feedback and helpful advice. I greatly appreciate his readiness to help at any time. I also want to thank my former fellow PhD students Dr. Tobias Seifarth and Niklas Lehne for many interesting discussions as well as Sebastian Blauth for his feedback during the final stages of this work.

Finally, I want to express my deepest gratitude for having a family, who always believed in me, always had an open ear and always gave me a place to come back home to and clear my mind. I also thank my girlfriend who, in the long period of writing and finishing up this thesis, must have thought my main relationship was with work, but kept on supporting me.

Contents

List of Figures

List of Tables

Introduction

Particulate media occurs in a wide variety of forms and on many different scales in nature and industry. It is not restricted to specific phases, since atomic constituents, liquid droplets or solid bodies can all be considered as *particles*. In this thesis we will restrict our attention towards particles which represent components of *granular materials*. By a granular material we denote a system of many solid particles which are small in comparison to the extent of the bulk, but far larger than the atomic scale. Such aggregations of solid particles can range from powders to sand, and from larger minerals to coffee beans and pills, which already suggests a wide spectrum of possible applications. We consider a numerical scheme to simulate the dynamics of such particle systems, which can be adapted to represent most of the above-mentioned scales. The number of particles which can be realized in these kind of approaches is however practically limited due to the scaling of computational cost with number of solid entities in the bulk. Therefore, in the case of fine sand and other powders, such as flour or ground coffee, it will only be possible to simulate rather small accumulations of this type, with the computational capacity which is currently available. Nevertheless, the simulation of granular media with larger grain sizes is highly relevant, for example, in different types of mixers [2, 89], for grinding in mineral processing [106], as well as screw-conveying mechanisms in chemical, food and pharmaceutical industry [107, 113]. Beyond these examples, where only the solid phase behavior needs to be simulated, applications like fluidized beds (used in the roasting of coffee beans and in many chemical reactors [46, 173]), the pneumatic conveying of different granular matter [91, 128] as well as cyclone devices (used for coal preparation and separation processes [18, 19]), suggest the importance of interactions between a fluid carrier phase and solid particles.

Traditionally, the dynamics of the carrier phase are resolved by a mesh-based numerical scheme such as finite volume or finite element methods. In these schemes, the creation of meshes for complex domains as well as the update of existing meshes due to moving parts or free surfaces is a difficult and time consuming task. This is where meshfree methods such as the *Finite Pointset Method* (FPM) can provide significant advantages. The FPM, which is a Lagrangian *generalized finite difference method* (GFDM), is based on irregularly spaced numerical points on which differential operators are approximated through a least-squares procedure. Due to its Lagrangian nature, points are moved with the continuum velocity, such that free surfaces and changes of the discretization caused by moving geometries are naturally represented. Additionally, most of the discretization process can be automated so that the generation of meshes as a manual pre-processing step is not necessary any more.

In this thesis, we introduce a Lagrangian-Lagrangian fluid-solid coupling in the FPM setting. Coupling, in this context, has to be understood as numerical coupling in the sense that the numerical schemes, which are introduced to describe different phases, are interconnected and embedded into an overarching solution procedure. While the rather general term *fluid-solid coupling* can also include *fluid-structure interactions* (FSI), we reiterate that this work is targeted towards the treatment of the above-mentioned reciprocal influence between a fluid phase and a large number of comparatively small solid entities which are immersed in this phase. We note that FSI problems have already been solved to great success within the finite pointset method [155] and were a driving force in the initial development of this numerical framework.

For the introduction of a numerical scheme describing the solid phase motion there are two major approaches which can be employed - a description of the macroscopic behavior of a large assembly of particles via continuous equations, on the one hand, or the tracking of individual particles and their interactions, on the other. Since the former approach can be solved with similar schemes on the same discretization as the fluid phase, it is often referred to as Eulerian-Eulerian coupling in the context of Eulerian mesh-based fluid solvers. On the other hand, the latter approach leads to a Lagrangian description in the solid phase and consequently to a Eulerian-Lagrangian coupling in the context of Eulerian mesh-based fluid solvers. This means that a Lagrangian framework needs to be maintained next to the Eulerian one. In conjunction with this, the necessity for mapping data between these two formulations arises. In the FPM, we are instead able to describe both phases in the same Lagrangian framework, thus resulting in a *Lagrangian-Lagrangian* coupling which provides us with useful synergies in terms of data structures and algorithms. The coupling approach we employ does not resolve individual particle surfaces by the fluid phase discretization, but relies on volume averaged equations in the fluid phase and the use of closure relations to model physical phenomena such as drag. While a mapping from fluid to solid phase is natural in the proposed framework, the reverse mapping will necessitate an averaging procedure which we analyze closely. In particular, due to the absence of an inherent definition of volume associated with entities of our numerical discretization, averaging schemes commonly used in Eulerian-Lagrangian algorithms can not be directly applied to our framework. We discuss ways of constructing such a notion of volume and present a procedure which circumvents the problem while taking the finite difference nature of the FPM into account. Then the remaining major component of our numerical coupling, the numerical solution of volume averaged equations in the fluid phase is presented. For this, we adapt the projection scheme of the FPM to the multiphase setting by accounting for changes in volume fraction and discuss the treatment of coupling source terms. The solid phase scheme, averaging procedure and fluid phase scheme are finally embedded into a coupled procedure with the possibility of taking several sub steps in the solid phase during each iteration of the coupled scheme. This *substepping procedure* reduces computational cost by preventing the strict time step restriction for the solid phase from severely limiting the fluid solver time step.

The structure of this thesis follows the narrative of this introduction. In particular, the first two chapters establish the mathematical model for the fluid phase and the generalized finite difference framework of the FPM. In the third chapter, the discussion of different solution procedures for the solid phase and types of coupling is picked up again and our choice of coupling scheme is properly motivated. The rest of this chapter will then be devoted to establishing the theoretical background of the chosen coupling scheme. Subsequently, all components of the coupled algorithm, which was introduced into the FPM, are detailed in chapter four. In particular, this includes the scheme for the solid phase motion, the averaging procedure and projection scheme for the fluid phase. Lastly, in chapter five, several test cases of increasing complexity are considered to analyze the capabilities of our coupled algorithm. In this context, we will also shed some light onto the changes to boundary conditions which have to be made in the fluid phase. We note that the test cases in chapter five only represent a subset of those problems which can be addressed with the presented algorithm and that the approach is not tailored towards one specific application, but we aimed to provide a general framework for multiphase problems of the discussed type within the FPM. Following this discussion of numerical results, this thesis is concluded by a summary of the presented work and we provide an outlook with suggestions for future extension.

Chapter 1

Foundations of fluid mechanics

The treatment of multiphase flows requires a good understanding of the mechanisms in single phase flows. This is why we first have to thoroughly derive the models employed in describing the relevant phases without the additional complications of interphase coupling. Along the way, we point out limitations and model assumptions that are inherent in the single phase descriptions and consequently also appear in the coupled models.

The fluid dynamical foundations discussed in the following can be found in numerous standard textbooks. We specifically adopt notations from and base our discussions on the lecture notes by Childress [16] and the book by Spurk and Aksel [142] (for which there is also an English version [141]) and the reader may consult these sources for more in-depth treatments.

A fluid phase is characterized by the fact that it deforms indefinitely under shear stresses and shows no resistance to deformation when the velocities vanish. In contrast to that, a solid material can support a shear stress in static equilibrium. As is evident from the above definition, we subsume both, liquids and gases, in the term *fluid* and refrain from the colloquial use of fluid as a synonym for liquid. The above material properties are consequences of microscopic structures, with the mean free paths of molecules being one important attribute. While liquids have a mean distance on the scale of a single molecule diameter, resulting in frequent interactions, a gas will have much larger mean free paths and interactions will occur comparatively rarely. For the continuum theory developed in the following, we consider observable quantities, such as pressure, that vary above these microscopic scales and are calculated as averages over volumes containing many molecules. Such a volume, called *fluid parcel*, is the smallest entity of matter that we consider. It needs to be small compared to the technical scale, relevant in the problems that should be modeled by the continuum theory, but has to be large enough to include sufficient numbers of molecules for a sensible definition of averages. In such a volume we define the important continuum property of fluid (*mass*) *density* ρ_f as the sum of molecule masses inside a parcel divided by its volume. With this, we can clarify an aspect of what a "sensible definition" is, namely that the volume is large enough that further increasing it would not alter the value of the density or any averaged quantity. On the other hand the continuum theory considers the fluid parcel as infinitely small compared to the fluid length scale of interest. This is why fluid parcels are treated as *material points* and the observable quantities as continuous functions in space and time. These assumptions can be considered valid for a wide variety of applications and form the basis for the treatment of fluid phases in this thesis, but it is important to note that they are by no means universally fulfilled. Examples where the continuum hypothesis is erroneous can be found by considering very small problems domains, extremely low gas densities or shock waves [6, 141].

1.1 Eulerian and Lagrangian descriptions

The movement of a single material point is given by its time dependent position vector $\boldsymbol{x}_\mathrm{f}(t) \in \mathbb{R}^d$ with $d \in \{1, 2, 3\}$. However, we want to describe the behavior of a finite volume of fluid composed of infinitely many material points. Let us assume that the considered fluid volume at time $t_0 \in \mathbb{R}_+ \coloneqq \{x \in \mathbb{R} \mid x \geq 0\}$ is given by the open set $S_0 \subset \mathbb{R}^d$. As the fluid moves, the material points follow their respective trajectories and at time $t \geq t_0$ the fluid will occupy a space $S_t \subset \mathbb{R}^d$. To describe the motion of the considered material volume we uniquely identify each material point through its original position $\boldsymbol{\xi} = \boldsymbol{x}_\mathrm{f}(t_0) \in S_0$ and write

$$\boldsymbol{x}_\mathrm{f} = \boldsymbol{X}(\boldsymbol{\xi}, t) \tag{1.1.1}$$

This formulation in terms of a reference position $\boldsymbol{\xi}$ and time t is called *Lagrangian coordinate* of a material point. If we consider a scalar quantity ψ (e.g. density or velocity) attached to the fluid, then in a Lagrangian description one examines the evolution in time of said quantity for a fixed set of material points. One can think of this as following the material points along their trajectories while sampling the quantity of interest. Despite certain advantages of a Lagrangian description, fluid properties are most commonly measured and prescribed at fixed points in space. Such a fixed point in space is potentially occupied by different material points as time progresses. Hence, in this so-called *Eulerian* description the observation of the scalar quantity can be written as a mapping of the form

$$(\boldsymbol{x}, t) \mapsto \psi(\boldsymbol{x}, t) \tag{1.1.2}$$

which means we observe ψ at a fixed point in space $\boldsymbol{x}$ and time t. In this formalism the Lagrangian description simply reads

$$(\boldsymbol{\xi}, t) \mapsto \psi(\boldsymbol{X}(\boldsymbol{\xi}, t), t) \tag{1.1.3}$$

With these observations we now want to define the Eulerian velocity field $\boldsymbol{u}_\mathrm{f}(\boldsymbol{x}, t)$. If we consider a specific material point, i.e. we keep $\boldsymbol{\xi}$ fixed, its velocity at time t is intuitively given by the time derivative of its Lagrangian coordinate, leading to the definition

$$\left.\frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}t}\right|_{\boldsymbol{\xi}} = \boldsymbol{u}_\mathrm{f}(\boldsymbol{X}(\boldsymbol{\xi}, t), t) \tag{1.1.4}$$

An implication of this relation is that, given a velocity field $\boldsymbol{u}_\mathrm{f}(\boldsymbol{x}, t)$, the Lagrangian coordinate of a material point with reference position $\boldsymbol{\xi}$ at time $t = 0$ can be obtained by solving the initial value problem

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}t} &= \boldsymbol{u}_\mathrm{f}(\boldsymbol{X}(t), t) \\ \boldsymbol{X}(0) &= \boldsymbol{\xi} \end{aligned} \tag{1.1.5}$$

We might also want to examine the change in the scalar quantity ψ while following a material point. This can be formulated as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\psi(\boldsymbol{X}(\boldsymbol{\xi}, t), t) &= \left[\boldsymbol{\nabla}\psi \cdot \frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}t} + \frac{\partial \psi}{\partial t}\right](\boldsymbol{X}(\boldsymbol{\xi}, t), t) \\ &= \left[\boldsymbol{\nabla}\psi \cdot \boldsymbol{u}_\mathrm{f} + \frac{\partial \psi}{\partial t}\right](\boldsymbol{X}(\boldsymbol{\xi}, t), t) \end{aligned} \tag{1.1.6}$$

Definition 1.1 *Let $\boldsymbol{u}_\mathrm{f} : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}$ be the Eulerian flow velocity field. The corresponding Eulerian operator*

$$\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \boldsymbol{u}_\mathrm{f} \cdot \boldsymbol{\nabla} \tag{1.1.7}$$

is called material or convective derivative.

This operator formulates the Lagrangian time derivative in terms of Eulerian quantities and will later be of particular importance. Specifically due to the fact that it includes the convective term $\boldsymbol{u}_\mathrm{f} \cdot \boldsymbol{\nabla}$ and can be used for the simplification of conservation laws in the fluid phase. To avoid any ambiguity and clarify our notation, we also want to detail how the material is applied to a vector field. In this case, the dot product is a product of the gradient tensor with the fluid velocity from the left, i.e.

$$\frac{\mathrm{D}\boldsymbol{v}}{\mathrm{D}t} = \frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{u}_\mathrm{f} \cdot \boldsymbol{\nabla}\boldsymbol{v} \tag{1.1.8}$$

with

$$\boldsymbol{u}_\mathrm{f} \cdot \boldsymbol{\nabla}\boldsymbol{v} = \left(\boldsymbol{u}_\mathrm{f} \cdot \boldsymbol{\nabla} v^{(1)}, \ldots, \boldsymbol{u}_\mathrm{f} \cdot \boldsymbol{\nabla} v^{(d)}\right)^T \tag{1.1.9}$$

We note that all vector and tensor notation within this work follows the tensor calculus described in Appendix A. The somewhat unusual choice of using superscripts for vector and tensor components is rooted in the notational difficulties which arise from the combination of multiphase flows and our numerical scheme.

1.2 Incompressibility

A fluid can be defined as being *incompressible* if for every material volume S_t the condition

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{S_t} 1 \,\mathrm{d}V = 0 \tag{1.2.1}$$

is fulfilled, i.e. the volume occupied by S_t does not change. It is, however, customary to use a different, more useful, definition of incompressibility in the context of fluid dynamics. To derive this, we consider the Jacobian of the Lagrangian coordinate $\boldsymbol{X}(\boldsymbol{\xi}, t)$ with respect to the reference position $\boldsymbol{\xi}$ at a fixed time t, i.e.

$$\mathrm{J}^{(i,j)}(\boldsymbol{\xi}, t) = \left.\frac{\partial X^{(i)}}{\partial \xi^{(j)}}\right|_t (\boldsymbol{\xi}, t) \tag{1.2.2}$$

Since the Lagrangian coordinate for fixed t can be considered as a map $\boldsymbol{X}(\cdot, t) : S_0 \to S_t$, we can rewrite the integral

$$\int_{S_t} 1 \,\mathrm{d}V = \int_{S_0} 1 |\det(\mathbf{J})| \,\mathrm{d}V_\xi \tag{1.2.3}$$

to observe that the incompressibility condition (1.2.1) is fulfilled for every volume $S_0 \subset \mathbb{R}^d$ if and only if $|\det(\mathbf{J})| = 1$ for all $t \in \mathbb{R}_+$. Note that we employ the common assumption that $\det(\mathbf{J}) > 0$ so that the map between S_0 and S_t is invertible and we can drop the absolute value in the following. For the invertible Jacobian $\mathbf{J}$, we use Jacobi's formula to obtain

$$\frac{\partial}{\partial t} \det(\mathbf{J}) = \mathrm{tr}\left(\mathbf{J}^{-1} \frac{\partial \mathbf{J}}{\partial t}\right) \det(\mathbf{J}) \tag{1.2.4}$$

By exchange of spatial and temporal differentiation one can verify that

$$\frac{\partial \mathbf{J}}{\partial t} = (\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f})(\boldsymbol{X}(\boldsymbol{\xi}, t), t)\mathbf{J}(\boldsymbol{\xi}, t) \tag{1.2.5}$$

Now, since $\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f}$ is square and $\mathbf{J}$ invertible, we further simplify

$$\mathrm{tr}\left(\mathbf{J}^{-1}\frac{\partial \mathbf{J}}{\partial t}\right) = \mathrm{tr}(\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f}) = \boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f} \tag{1.2.6}$$

which finally leads to

$$\frac{\partial}{\partial t}\det(\mathbf{J})(\boldsymbol{\xi}, t) = [\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f}(\boldsymbol{X}(\boldsymbol{\xi}, t), t)]\det(\mathbf{J})(\boldsymbol{\xi}, t) \tag{1.2.7}$$

Together with $\det(\mathbf{J}) = 1$ we obtain the condition that is typically used to define incompressible flows in Eulerian variables and stated in the following theorem.

Theorem 1.1 *In Eulerian description, a fluid covering a flow domain represented by the open set $\Omega \subset \mathbb{R}^d$ is incompressible if and only if*

$$\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f} = 0 \tag{1.2.8}$$

for every $(\boldsymbol{x}, t) \in \Omega \times \mathbb{R}_+$.

1.3 Convection theorem

Now that we have laid the necessary groundwork, we use fundamental conservation principles to derive equations that describe the motion of fluids in terms of the aforementioned averaged quantities. To this end, let us consider a region of material points $S_t \subset \mathbb{R}^d$ and a scalar function $\psi : \mathbb{R}^d \times \mathbb{R}_0^+ \to \mathbb{R}$. Using the integral transformation to the domain of representative points S_0, we can derive

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{S_t} \psi(\boldsymbol{x}, t)\,\mathrm{d}V_x = \frac{\mathrm{d}}{\mathrm{d}t}\int_{S_0} \psi(\mathbf{X}(\boldsymbol{\xi}, t), t)\det(\mathbf{J})\,\mathrm{d}V_\xi \tag{1.3.1}$$

$$= \int_{S_0} \det(\mathbf{J})\frac{\mathrm{d}}{\mathrm{d}t}\psi(\mathbf{X}(\boldsymbol{\xi}, t), t)\,\mathrm{d}V_\xi + \int_{S_0} \psi(\mathbf{X}(\boldsymbol{\xi}, t), t)\frac{\partial}{\partial t}\det(\mathbf{J})\,\mathrm{d}V_\xi \tag{1.3.2}$$

Following (1.1.6) and (1.1.7), the first integral on the right-hand side can be rewritten in the material derivative notation, while for the second term we employ (1.2.7). From this we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{S_t} \psi(\boldsymbol{x}, t)\,\mathrm{d}V_x = \int_{S_0} \left[\frac{\mathrm{D}\psi}{\mathrm{D}t} + \psi(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})\right]\det(\mathbf{J})\,\mathrm{d}V_\xi \tag{1.3.3}$$

which, by transforming back to S_t, results in

Theorem 1.2 *Let $S_t \subset \mathbb{R}^d$ be a material volume of fluid identified by the reference positions $S_0 \subset \mathbb{R}^d$ and the Eulerian velocity field $\boldsymbol{u}_\mathrm{f}$. If we assume that both, the velocity and a scalar function ψ, are sufficiently smooth, the relation*

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{S_t} \psi(\boldsymbol{x}, t)\,\mathrm{d}V_x = \int_{S_t} \frac{\mathrm{D}\psi}{\mathrm{D}t} + \psi(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})\,\mathrm{d}V_x \tag{1.3.4}$$

holds and is called the convection theorem. When considered in the equivalent form

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{S_t}\psi(\boldsymbol{x},t)\,\mathrm{d}V_x=\int_{S_t}\frac{\partial\psi}{\partial t}+\boldsymbol{\nabla}\cdot(\psi\boldsymbol{u}_\mathrm{f})\,\mathrm{d}V_x \tag{1.3.5}$$

it is often referred to as Reynolds transport theorem.

1.4 Conservation of mass

The mass of a material volume with initial configuration $S_0\subset\mathbb{R}^d$ and positions $S_t\subset\mathbb{R}^d$ at time $t\in\mathbb{R}_+$ can be calculated by integrating the (mass) density ρ_f, i.e.

$$m(S_0,t)=\int_{S_t}\rho_\mathrm{f}(\boldsymbol{x},t)\,\mathrm{d}V_x \tag{1.4.1}$$

When no material is added or removed from the computational domain, there is no possibility for the mass to change with respect to time, which together with Reynolds transport theorem can be formulated as the condition

$$0=\frac{\mathrm{d}m}{\mathrm{d}t}=\int_{S_t}\frac{\partial\rho_\mathrm{f}}{\partial t}+\boldsymbol{\nabla}\cdot(\rho_\mathrm{f}\boldsymbol{u}_\mathrm{f})\,\mathrm{d}V_x \tag{1.4.2}$$

We observe that the integral vanishes for any material volume, from which can conclude that the integrand evaluates to zero point-wise inside the whole domain covered by the fluid, finally leading us to

$$\frac{\partial\rho_\mathrm{f}}{\partial t}+\boldsymbol{\nabla}\cdot(\rho_\mathrm{f}\boldsymbol{u}_\mathrm{f})=0 \tag{1.4.3}$$

This important physical law is called *mass conservation equation* or *continuity equation.* By applying product rule to the divergence term we can reformulate the condition to obtain

$$\frac{\mathrm{D}\rho_\mathrm{f}}{\mathrm{D}t}+\rho_\mathrm{f}(\boldsymbol{\nabla}\cdot\boldsymbol{u}_\mathrm{f})=0 \tag{1.4.4}$$

which is the Lagrangian formulation of mass conservation that we make extensive use of when we consider the chosen numerical scheme for computing fluid flows. If we recall that an incompressible fluid has vanishing velocity divergence, we observe that the above conservation equation reduces to

$$\frac{\mathrm{D}\rho_\mathrm{f}}{\mathrm{D}t}=0 \tag{1.4.5}$$

which means that incompressibility also implies constant density along the paths of material points.

Remark 1.1 *Note that mass addition or removal can be included by a function $q(\boldsymbol{x},t)$ which has the dimensions of mass per unit volume and time. This function fulfills*

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_R\rho_\mathrm{f}(\boldsymbol{x},t)\,\mathrm{d}V_x=\int_R q(\boldsymbol{x},t)\,\mathrm{d}V_x \tag{1.4.6}$$

for a fixed subset $R\subset\mathbb{R}^d$ of the flow domain and one can show [16] that this leads to

$$\frac{\mathrm{D}\rho_\mathrm{f}}{\mathrm{D}t}+\rho_\mathrm{f}(\boldsymbol{\nabla}\cdot\boldsymbol{u}_\mathrm{f})=q \tag{1.4.7}$$

In the context of multiphase flows, one can think of cases such as the dissolution of solids in a fluid phase, where such a term will arise. In the context of this work, we do, however, restrict ourselves to settings without mass transfer between phases or other mass sources and sinks in the fluid phase.

1.5 Conservation of momentum

In addition to the density, we also need to determine the velocity components from conservation laws. Since the mass conservation on its own only provides us with a single equation for the $d+1$ unknowns $\rho_\mathrm{f}, u_\mathrm{f}^{(1)}, \dots, u_\mathrm{f}^{(d)}$, it is necessary to derive additional equations to fully describe the fluid flow. To do so, we apply Newton's second law to a material volume for which we define the total momentum by

$$\boldsymbol{p}(S_0, t) = \int_{S_t} \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \,\mathrm{d}V_x \tag{1.5.1}$$

With total forces $\boldsymbol{f}(S_0, t)$ acting on the material volume we obtain

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \boldsymbol{f} \tag{1.5.2}$$

We can split $\boldsymbol{f}$ into contributions from body forces and forces acting on the surface ∂S_t such that

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{S_t} \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \,\mathrm{d}V_x = \int_{S_t} \rho_\mathrm{f} \boldsymbol{k} \,\mathrm{d}V_x + \int_{\partial S_t} \boldsymbol{t} \,\mathrm{d}S_x \tag{1.5.3}$$

where $\boldsymbol{k}$ denotes the *body force density* and $\boldsymbol{t}$ denotes the *stress vector*. While the body force density subsumes terms such as magnetic forces or gravity, the stress vector represents the force onto the material surface due to surrounding fluid. The stress vector can be shown [141, 142] to have a linear relation $\boldsymbol{t} = \boldsymbol{\sigma}_\mathrm{f} \boldsymbol{n}$ to the surface normal, where $\boldsymbol{\sigma}_\mathrm{f}$ denotes the second order tensor called *stress tensor*. Applying the divergence theorem we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{S_t} \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \,\mathrm{d}V_x = \int_{S_t} \rho_\mathrm{f} \boldsymbol{k} \,\mathrm{d}V_x + \int_{S_t} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_\mathrm{f} \,\mathrm{d}V_x \tag{1.5.4}$$

If we apply Reynolds theorem component-wise to $\rho_\mathrm{f} u_\mathrm{f}^{(i)}$ we encounter the terms

$$\boldsymbol{\nabla} \cdot \left(\rho_\mathrm{f} u_\mathrm{f}^{(i)} \boldsymbol{u}_\mathrm{f} \right), \quad i = 1, \dots, d \tag{1.5.5}$$

which can be rewritten into the vectorial form

$$\boldsymbol{\nabla} \cdot (\boldsymbol{u}_\mathrm{f} \otimes \boldsymbol{u}_\mathrm{f}) = \begin{bmatrix} \boldsymbol{\nabla} \cdot \left(\rho_\mathrm{f} u_\mathrm{f}^{(1)} \boldsymbol{u}_\mathrm{f} \right) \\ \vdots \\ \boldsymbol{\nabla} \cdot \left(\rho_\mathrm{f} u_\mathrm{f}^{(d)} \boldsymbol{u}_\mathrm{f} \right) \end{bmatrix} \tag{1.5.6}$$

where the outer product $\otimes$ and divergence notation for second rank tensors are properly defined in Appendix A. Putting everything together, we finally obtain the integral equation

$$\int_{S_t} \frac{\partial \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \otimes \boldsymbol{u}_\mathrm{f}) \,\mathrm{d}V_x = \int_{S_t} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_\mathrm{f} + \rho_\mathrm{f} \boldsymbol{k} \,\mathrm{d}V_x \tag{1.5.7}$$

which, with the same arguments as before, leads to the point-wise *momentum conservation equations* in Eulerian form

$$\frac{\partial \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \otimes \boldsymbol{u}_\mathrm{f}) = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_\mathrm{f} + \rho_\mathrm{f} \boldsymbol{k} \tag{1.5.8}$$

If we split the partial time derivative on the left-hand side by product rule and use the identity

$$\boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{v}) = \boldsymbol{v} \cdot \boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \cdot \boldsymbol{v}) \boldsymbol{u} \tag{1.5.9}$$

with $\boldsymbol{u} = \boldsymbol{u}_\mathrm{f}$ and $\boldsymbol{v} = \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}$, we obtain

$$\frac{\partial \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \otimes \boldsymbol{u}_\mathrm{f}) = \left[\frac{\partial \rho_\mathrm{f}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}) \right] \boldsymbol{u}_\mathrm{f} + \rho_\mathrm{f} \left[\frac{\partial \boldsymbol{u}_\mathrm{f}}{\partial t} + \boldsymbol{u}_\mathrm{f} \cdot \boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f} \right] \tag{1.5.10}$$

We observe that the first summand vanishes due to the mass conservation equation (1.4.3) and that the second term equals the material derivative of velocity. Thus, we can write down the equivalent Lagrangian form of momentum conservation as

$$\rho_\mathrm{f} \frac{\mathrm{D} \boldsymbol{u}_\mathrm{f}}{\mathrm{D} t} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_\mathrm{f} + \rho_\mathrm{f} \boldsymbol{k} \tag{1.5.11}$$

This form of the equation is sometimes also referred to as the *Cauchy equation* and it is applicable to every continuous phase irrespective of the specific material properties. Since the body force term $\boldsymbol{k}$ can be considered a given quantity and the second rank stress tensor is symmetric, we added $d(d+1)/2$ variables while only adding d equations. From this fact it is already clear that the conservation equations alone are not sufficient to describe the fluid behavior. As a first step in defining the necessary closure relation, it is customary to split off a pure normal stress tensor whose identical components take the value of *(thermodynamical) pressure* p, i.e.

$$\boldsymbol{\sigma}_\mathrm{f} = -p \mathbb{1} + \boldsymbol{\tau}_\mathrm{f} \tag{1.5.12}$$

The remaining, not necessarily deviatoric, tensor $\boldsymbol{\tau}_\mathrm{f}$ is called *viscous stress tensor*. It describes the stresses due to friction inside the fluid and defines the fluid rheology. Due to the fact that friction can only arise when there are parts that move relative to each other, we can directly conclude that $\boldsymbol{\tau}_\mathrm{f}$ will in some way depend on spatial derivatives of the fluid velocity. The specific form of the tensor can, however, vary strongly on a per-material basis and is enclosed in so-called *constitutive equations*. They form the transition from universal conservation laws to material-specific models.

The easiest closure is given by the assumption of inviscid flow, i.e. the frictional stresses within the fluid are neglected and the stress tensor reduces to only a pressure component. This closure results in the well-known *Euler equations* for momentum conservation. While this might seem like a crude approximation, there is a wide variety of flow configurations where these equations can be sufficient to describe the flow behavior. It's applicability is often decided based on a dimensionless characteristic number, the *Reynolds number*, which we introduce later. But before, we want to discuss a slightly more complex and widely used closure.

This closure considers the internal friction and is obtained by assuming that the stress is Galilean invariant and fulfills Hooke's law. This means that the stresses only depend on the gradients and not the flow velocity itself and does so linearly, i.e.

$$\boldsymbol{\tau}_\mathrm{f} = \mathbf{C} \colon (\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f}) \tag{1.5.13}$$

where $\mathbf{C}$ is a fourth order tensor. Further assuming that the fluid is isotropic and homogeneous we obtain the viscous stress tensor of a *Newtonian fluid* (cf. [87, 141, 142])

$$\boldsymbol{\tau}_\mathrm{f} = \lambda \operatorname{tr}(\mathbf{S}) \mathbb{1} + 2\mu \mathbf{S} \tag{1.5.14}$$

with λ denoting the first Lamé parameter, μ the *dynamic viscosity* and $\mathbf{S}$ the *linearized strain tensor*. The matrix representation of $\mathbf{S}$ is obtained by splitting the velocity Jacobian into a symmetric and skew-symmetric part, i.e.

$$\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f} = \mathbf{S} + \boldsymbol{\Omega}, \quad \mathbf{S} = \frac{1}{2}\Big(\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f} + (\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f})^T\Big), \quad \boldsymbol{\Omega} = \frac{1}{2}\Big(\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f} - (\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f})^T\Big) \tag{1.5.15}$$

such that the viscous stress tensor takes the form

$$\boldsymbol{\tau}_\mathrm{f} = \lambda(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})\mathbb{1} + \mu\Big(\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f} + (\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f})^T\Big) \tag{1.5.16}$$

Although non-Newtonian fluids are by no means rare in industrial applications, the class of Newtonian fluids includes many practically important liquids and gases such as water, air and different kinds of oils.Therefore we restrict ourselves to these fluids to simplify the single phase models while still keeping a foundation which is highly relevant in applications.

Another formulation of the viscous stress tensor, which is commonly used, can be obtained by splitting the tensor into an isotropic and deviatoric part. From $\mathrm{tr}(2\mathbf{S}) = 2(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})$ and $\mathrm{tr}((\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})\mathbb{1}) = 3(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})$ we can easily see that this is fulfilled by

$$\boldsymbol{\tau}_\mathrm{f} = \zeta(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})\mathbb{1} + \mu\Big(\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f} + (\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f})^T - \frac{2}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})\mathbb{1}\Big) \tag{1.5.17}$$

with $\zeta = \lambda + \frac{2}{3}\mu$ denoting the *volume viscosity*, sometimes also called *second viscosity*. In particular for liquids, it has been extensively validated that one can, in many cases, neglect the first term in (1.5.17) such that the viscous stress tensor is equal to the deviatoric part of the full stress tensor $\boldsymbol{\sigma}_\mathrm{f}$ (cf. [141, 142]). This assumption is often identified with Stokes' hypothesis which states that the volume viscosity vanishes, i.e.

$$\zeta = \lambda + \frac{2}{3}\mu = 0 \tag{1.5.18}$$

However, as described in [13], while the simplified equations have shown to be valid, the assumption $\zeta = 0$ is for many liquids and gases in contradiction to experimental and theoretical findings. They further claim that the simplification is still valid since the more general criterion

$$|\zeta(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})| \ll p \tag{1.5.19}$$

is fulfilled in all these cases and the whole term $\zeta(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f})$ can be neglected in the stress tensor due to its negligible magnitude in comparison to the thermodynamic pressure. We note that (1.5.17), with the volume viscosity term neglected, is used for the viscous stresses throughout this thesis.

The parameters λ, ζ, μ are, not necessarily constant, material properties which can be considered given when solving the presented set of equations. So the closure relation obtained from (1.5.12) and (1.5.16) introduces an additional scalar quantity p while expressing the rest of the tensor in terms of the existing variable $\boldsymbol{u}_\mathrm{f}$. For an incompressible fluid we have already seen that the mass conservation implies constant density along material point trajectories. We can thus consider the density as a given quantity since we can always trace back from the considered Eulerian point $\boldsymbol{x}$ to the reference point $\boldsymbol{\xi}$, where the value is assumed to be known. With one variable crossed out, we have a closed set of equations consisting of the incompressibility condition and the momentum conservation with the simplified viscous stress

$$\boldsymbol{\tau}_\mathrm{f} = \mu\Big(\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f} + (\boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f})^T\Big) \tag{1.5.20}$$

The most commonly referenced version of this closed set can be obtained by assuming spatially non-varying viscosities ζ and μ. Then the divergence of the full stress tensor reads

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_{\mathrm{f}} = -\boldsymbol{\nabla} p + \mu \nabla^2 \boldsymbol{u}_{\mathrm{f}} + (\lambda + \mu) \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{u}_{\mathrm{f}}) \tag{1.5.21}$$

where we used

$$\boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} \boldsymbol{u}_{\mathrm{f}}) = \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{u}_{\mathrm{f}}), \quad \boldsymbol{\nabla} \cdot \left((\boldsymbol{\nabla} \boldsymbol{u}_{\mathrm{f}})^T \right) = \nabla^2 \boldsymbol{u}_{\mathrm{f}} \tag{1.5.22}$$

For an incompressible fluid, the last term on the right-hand side of (1.5.21) vanishes and we obtain the closed set of equations

$$\boldsymbol{\nabla} \cdot \boldsymbol{u}_{\mathrm{f}} = 0 \tag{1.5.23}$$

$$\rho_{\mathrm{f}} \left[\frac{\partial \boldsymbol{u}_{\mathrm{f}}}{\partial t} + \boldsymbol{u}_{\mathrm{f}} \cdot \boldsymbol{\nabla} \boldsymbol{u}_{\mathrm{f}} \right] = -\boldsymbol{\nabla} p + \mu \nabla^2 \boldsymbol{u}_{\mathrm{f}} + \rho_{\mathrm{f}} \boldsymbol{k} \tag{1.5.24}$$

which are commonly called *incompressible Navier-Stokes equations*. Note that it is customary to divide both sides of the momentum equation by the density and define the *kinematic viscosity*

$$\eta = \frac{\mu}{\rho_{\mathrm{f}}} \tag{1.5.25}$$

We further observe that the pressure only occurs through its gradient, such that, without boundary conditions, it is always only defined up to an additive constant.

If we now consider compressible fluids, the set of equations consisting of mass conservation (1.4.3) and momentum conservation (1.5.8) with Newtonian stress (1.5.16) is not closed. This problem is typically solved in fluid dynamics by introducing an additional energy conservation equation and an equation of state. While we are faced with varying density flows in the multiphase setting, these variations are only modifications of the fluid phase density due to the presence of solid entities. In particular, we neither have to rely on the energy equation nor on an equation of state, which is why details on these topics are omitted at this point.

This concludes our discussion on the fundamental equations describing the fluid flow in a single-phase setting. Below, we have provided two additional remarks which have been omitted before to improve readability.

Remark 1.2 *It is customary to consider the Navier-Stokes equations in a non-dimensionalized form in order to obtain characteristic numbers, which help to identify similar flow behavior that may occur on completely different absolute scales. To define such a number, we assume that the flow has a characteristic length scale L and velocity U and define*

$$\tilde{\boldsymbol{x}} = \frac{\boldsymbol{x}}{L}, \quad \tilde{\boldsymbol{u}}_{\mathrm{f}} = \frac{\boldsymbol{u}_{\mathrm{f}}}{U}, \quad \tilde{t} = \frac{t}{L/U}, \quad \tilde{p} = \frac{p}{\rho_{\mathrm{f}} U^2}, \quad \tilde{\boldsymbol{\nabla}} = \left[\frac{\partial}{\partial \tilde{x}^{(1)}} \cdots \frac{\partial}{\partial \tilde{x}^{(d)}} \right]^T$$

Omitting body forces for now, the momentum equation (1.5.24) *can be rewritten as*

$$\frac{\partial \tilde{\boldsymbol{u}}_{\mathrm{f}}}{\partial \tilde{t}} + \tilde{\boldsymbol{u}}_{\mathrm{f}} \cdot \tilde{\boldsymbol{\nabla}} \tilde{\boldsymbol{u}}_{\mathrm{f}} = -\tilde{\boldsymbol{\nabla}} \tilde{p} + \frac{1}{Re} \tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{\nabla}} \tilde{\boldsymbol{u}}_{\mathrm{f}} \tag{1.5.26}$$

where

$$Re = \frac{\rho_{\mathrm{f}} U L}{\mu} \tag{1.5.27}$$

denotes the aforementioned Reynolds number. This quantity will prove to be important at a later point in this thesis, when we consider expressions that describe the interaction between

fluid and solid phases. We further note that for large Reynolds numbers the viscous term clearly looses significance and in these cases it can often be a valid approximation to use the Euler equations, which we previously obtained from the assumption of a frictionless fluid.

Chapter 2

The Finite Pointset Method

Before we make the transition from single phase flow to the main goal of this work, the coupled simulation of fluid-solid multiphase flow, we want to introduce the numerical framework in which we intend to solve this problem. A general understanding of this background will simplify the constructive discussion on suitable coupling approaches for the targeted application within the next chapter. The method we employ has been referred to as *Finite Pointset Method* (FPM) [40, 73] and is realized in the commercial software *MESHFREE*[1]. It belongs to the group of meshfree *generalized finite difference methods* (GFDMs).

The idea of GFDMs has first been proposed by Liszka and Orkisz [93], who generalized the construction of classical finite difference stencils from ordered grids to domains discretized by disordered points. The approach relies on the use of weighted least squares approximations to calculate stencils which minimize the error resulting from local Taylor expansions.

In the finite pointset method, these stencils are used to solve partial differential equations in strong form on points which move with the continuum velocity. This Lagrangian approach has been developed in order to improve on the boundary handling and accuracy of other Lagrangian meshfree methods such as the *smoothed particle hydrodynamics* (SPH) method [83]. An important distinguishing factor is that points are strictly numerical entities instead of mass particles, which allows for the addition and deletion of points in order to improve accuracy and stability.

Since its inception, the finite pointset method (FPM) has shown good results for a range of different problems including geomechanics [105, 111], automotive engineering [73, 155], chemical engineering [40] and process design [122, 159]. In addition to MESHFREE, it also serves as the numerical basis for *NOGRID*[2] and has been incorporated into *VPS* (formerly *PAM-CRASH*) [155], both of which are commercial software tools.

While the approach is applicable to a wide range of continuum mechanical problems, we will, in light of the targeted application and for the sake of simplicity, think of the continuum as a fluid phase. To introduce the differential operators in our GFDM framework, we first want to discuss how the domain is discretized.

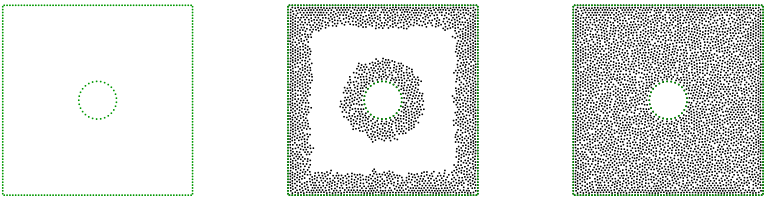
2.1 Spatial discretization and neighboring points

Let us assume that at a time $t \in \mathbb{R}_+$ the volume $\Omega(t) \in \mathbb{R}^d$, $d \in \{2, 3\}$, is occupied by the fluid phase. In order to solve the differential equations, such as the ones derived in Chapter 1, we discretize this computational domain by a set of numerical points with positions

$$\boldsymbol{x}_i \in \mathbb{R}^d, \quad i \in \{1, \dots, N\} \tag{2.1.1}$$

In contrast to the typical procedure in mesh-based methods, this set of numerical points, often referred to as *point cloud*, is constructed through automated filling of the domain. This

[1] https://www.meshfree.eu
[2] https://www.nogrid.com



(a) Initial points on boundary. **(b)** Result after a few filling steps. **(c)** End result of filling.

Figure 2.1: Successive stages of the initial point filling procedure in the FPM.

is done by first placing points on boundary elements, followed by recursive application of a hole filling algorithm (see Figure 2.1). Details on this algorithm can be found in [40] and will not be discussed here. For the purpose of this introduction it suffices to know that the point cloud is constructed in such as way as to ensure that accurate differential operators can be calculated at each point position.

Analogous to classical finite differences, the differential operator approximations at a point $\boldsymbol{x}_i$ are based on adjacent points. While the meaning of "adjacent" is quite clear in the mesh-based finite difference setting, it needs clarification in our setup. Within our framework, the differential operator approximations at $\boldsymbol{x}_i$ are based on all *neighboring points* $\boldsymbol{x}_j$ with index j in the set

$$\mathcal{N}_i \coloneqq \{j \mid j \in \{1, \dots, N\},\ \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq h\} \tag{2.1.2}$$

where h denotes the so-called *smoothing length*. In the following we also often refer to this set of indices, or the corresponding set of points for that matter, as the *neighborhood* of a point. We further note that the above definition includes the point $\boldsymbol{x}_i$ itself. In addition to the above notation, we also repeatedly make use of the subscript notation

$$\mathcal{N}_i = \{j_1, \dots, j_{n_i}\} \tag{2.1.3}$$

where n_i denotes the number of neighbors for particle i. In fact, this is not only a construct for notational convenience, but also reflects the implementation structure, where each point has a list of neighbors with *local indices* $k \in \{1, \dots, n_i\}$ which are mapped to *global indices* j_k when any data is retrieved or modified. For the sake of simplicity we assume that the list of neighbors is ordered with ascending distance to the center point. In particular, we make use of the fact that $j_1 = i$ in the above set of indices $\mathcal{N}_i$.

Examples of such point neighborhoods, both, close to and far away from the boundary of the domain depicted in Figure 2.1, are shown in Figure 2.2. In particular, we observe that, while the number of points in $\mathcal{N}_i$ reduces close to boundaries, it still significantly larger than in many classical FDMs. This is one of the aforementioned ways to ensure accurate differential operators at the center point, since it helps relax the requirements on regularity of the point cloud. Typically, for points far away from the boundary, the size of neighborhoods n_i is in the range of 15-20 points for two-dimensional problems and 45-50 in 3D.

We further note that the point clouds are constructed in such a way that the distance to the nearest neighbor fulfills

$$r_{\min} h \leq \min_{\substack{j \in \mathcal{N}_i \\ j \neq i}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq r_{\text{hole}} h \tag{2.1.4}$$

for some parameters $r_{\min},\ r_{\text{hole}} \in (0, 1)$. Taking $r_{\min} > 0$ ensures that points are not too close to each other while $r_{\text{hole}} < 1$ ensures that no holes of radius $r_{\text{hole}} h_i$ may form within

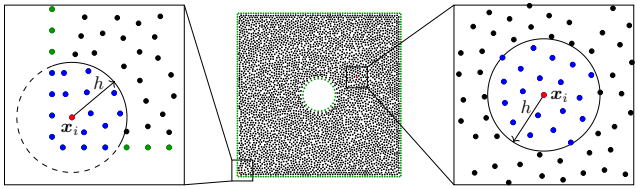


Figure 2.2: Left: Neighbor points (blue) for a FPM point (red) close to the boundary (green). Middle: Complete discretization of computational domain (see Figure 2.1). Right: Neighbor points (blue) for a FPM point (red) far from the boundary.

the computational domain. From a wealth of experience with our implementation of the numerical method, values of approximately $r_{\text{min}} = 0.15$ and $r_{\text{hole}} = 0.4$ have been found to be appropriate. Since these two parameters are typically kept fixed, we observe a direct connection between inter-point distances and the smoothing length h_i, which is consequently taken as the primary parameter of spatial discretization and often viewed as an analogue to the grid spacing in mesh-based methods. For an easier introduction of the above definitions, we omitted the fact that the smoothing length and thus the numerical resolution can be varied on a point basis, i.e. there is a value $h_i = h(\boldsymbol{x}_i(t), t)$ attached to each point. Both, the spatial and temporal variation of smoothing length, can be necessary to capture complex physical phenomena and can also be used to coarsen the resolution to reduce computational load in regions with simple flow behavior. In cases where such a varying smoothing length is employed, we replace h by an average smoothing length h_{ij} between the two points. Two possible definitions of this value are given by

$$h_{ij} = \frac{1}{2}(h_i + h_j) \tag{2.1.5}$$

or

$$h_{ij}^2 = \frac{2}{\frac{1}{h_i^2} + \frac{1}{h_j^2}} \tag{2.1.6}$$

where we typically resort to the latter definition. These definitions ensure that points are in each other's neighborhood and that distances scaled by smoothing length are symmetric.

With a discretized domain, such as the one in Figure 2.1c, and neighbor relations being established, we now want to provide a concise introduction to the construction of approximations to differential operators within the finite pointset method.

2.2 Construction of differential operators

Let us consider the generalized operator ∂_* which can represent a function value approximation, a partial derivative with respect to any of the spatial dimensions or other operators such as the laplacian. Our goal is to determine coefficients c_{ij}^* which provide an approximation

$$\partial_* f(\boldsymbol{x}_i) \approx \sum_{j \in \mathcal{N}_i} c_{ij}^* f(\boldsymbol{x}_j) \tag{2.2.1}$$

based on function values at neighboring points. It is important to note that these coefficients are independent of the function f, which means that, for a given point configuration, they only need to be determined once and can then be applied to various quantities in the underlying differential equation. In the following we refer to the vector of coefficients

$$\boldsymbol{c}_i^* = \left(c_{ij_1}, \ldots, c_{ij_{n_i}}\right)^T \in \mathbb{R}^{n_i} \tag{2.2.2}$$

as a *stencil* and denote the vector of all function values at neighboring points by

$$\boldsymbol{f}_{\mathcal{N}_i} = \Big(f(\boldsymbol{x}_{j_1}), \dots, f(\boldsymbol{x}_{j_{n_i}})\Big)^T \in \mathbb{R}^{n_i} \tag{2.2.3}$$

With this notation we can rewrite (2.2.1) as

$$\partial_* f(\boldsymbol{x}_i) \approx \boldsymbol{c}_i^* \cdot \boldsymbol{f}_{\mathcal{N}_i} \tag{2.2.4}$$

As noted above, the operators represented by the "$*$"-symbol include, but are not limited to, the approximation or smoothing of function values

$$f(\boldsymbol{x}_i) \approx \sum_{j \in \mathcal{N}_i} c_{ij}^0 f(\boldsymbol{x}_j) = \boldsymbol{c}_i^0 \cdot \boldsymbol{f}_{\mathcal{N}_i} \tag{2.2.5}$$

and the approximation of first order partial derivatives such as

$$\frac{\partial f}{\partial x}(\boldsymbol{x}_i) \approx \sum_{j \in \mathcal{N}_i} c_{ij}^x f(\boldsymbol{x}_j) = \boldsymbol{c}_i^x \cdot \boldsymbol{f}_{\mathcal{N}_i} \tag{2.2.6}$$

Depending on the form of differential equation and involved higher order derivatives, the operators can take more general, numerically advantageous, forms. In the case of the incompressible Navier-Stokes equations (1.5.23)-(1.5.24), the necessary stencils in $\mathbb{R}^3$ are

$$\boldsymbol{c}_i^0,\ \boldsymbol{c}_i^x,\ \boldsymbol{c}_i^y,\ \boldsymbol{c}_i^z,\ \boldsymbol{c}_i^{\nabla^2} \qquad i = 1, \dots, N \tag{2.2.7}$$

which correspond to function evaluation, the partial derivatives w.r.t. x, y, z, as well as the Laplacian. Once these stencils are determined. they are used analogously to classical finite difference methods to assemble sparse linear systems within the solution procedure of the PDE.

There are two different ways to derive these stencils which have been shown to be equivalent [144]. On the one hand, we can can follow the classical finite difference approach of considering Taylor expansions up to the required order and then apply a weighted least squares approach to minimize the errors resulting from the representation by a finite number of terms (cf. [40, 146]). This is also the approach which was proposed in the original work on generalized finite difference methods by Lizska and Orkisz [93]. The alternative approach, which we describe here in more detail, is based on the idea of prescribing the result that stencils should provide when applied to given test functions (cf. [73, 136, 147]).

The test functions we employ here are the monomials

$$\mathcal{M}_i = \left\{ \prod_{k=1}^{d} \frac{1}{p_k!} \Big(x^{(k)} - x_i^{(k)}\Big)^{p_k} \;\middle|\; p_k \in \mathbb{N}_0,\ \sum_{k=1}^{d} p_k \leq \mathrm{P} \right\} \tag{2.2.8}$$

in $\mathbb{R}^d$ and with maximum order P. We recall from Chapter 1 that superscripts in parenthesis are used to denote vector components, so that the k-th component of $\boldsymbol{x}$ is given by $x^{(k)}$. The reasoning behind this notation also becomes apparent now, since we have to distinguish between point indices, vector components and exponents. To derive stencils of a desired order, we demand that an application of those stencils to all of the above monomials up to that order leads to the exact result, i.e.

$$\partial_* f(\boldsymbol{x}_i) = \boldsymbol{c}_i^* \cdot \boldsymbol{f}_{\mathcal{N}_i}, \quad \forall f \in \mathcal{M}_i \tag{2.2.9}$$

Clearly, the same holds true for all polynomials up to the maximum oder P, since $\mathcal{M}$ functions as a basis for the space of polynomials. To collect all these conditions within a matrix-vector formulation, we first write the set of monomials as

$$\mathcal{M}_i = \{f_1, f_2, \dots, f_m\} \tag{2.2.10}$$

where m is the number of elements in $\mathcal{M}_i$ given by

$$m = \binom{d+\mathrm{P}}{d} = \frac{(d+\mathrm{P})!}{\mathrm{P}!d!} \tag{2.2.11}$$

For a given differential operator ∂_* we collect the left-hand sides of (2.2.9), i.e. the exact results of applying the continuous operator to the monomials at the center point $\boldsymbol{x}_i$, in

$$\boldsymbol{b}^* = (\partial_* f_1(\boldsymbol{x}_i), \partial_* f_2(\boldsymbol{x}_i), \ldots, \partial_* f_m(\boldsymbol{x}_i))^T \in \mathbb{R}^m \tag{2.2.12}$$

This vector provides the right-hand side in our matrix-vector formulation while the stencil $\boldsymbol{c}_i^* \in \mathbb{R}^{n_i}$ evidently is the sought-for variable. To construct the remaining matrix $\mathbf{M}_i \in \mathbb{R}^{m \times n_i}$ based on the $\boldsymbol{f}_{\mathcal{N}_i}$ in (2.2.9), we define

$$\mathrm{M}_i^j = (f_1(\boldsymbol{x}_j), f_2(\boldsymbol{x}_j), \ldots, f_m(\boldsymbol{x}_j))^T \in \mathbb{R}^m \tag{2.2.13}$$

for $j \in \mathcal{N}_i = \{j_1, \ldots, j_{n_i}\}$ so that we can simply write

$$\mathbf{M}_i = \begin{bmatrix} \mathrm{M}_i^{j_1} & \ldots & \mathrm{M}_i^{j_{n_i}} \end{bmatrix} \in \mathbb{R}^{m \times n_i} \tag{2.2.14}$$

With these notations established, the so-called *consistency conditions* can be written in the succinct form

$$\mathbf{M}_i \boldsymbol{c}_i^* = \boldsymbol{b}^* \tag{2.2.15}$$

In general we have more neighboring points than monomials such that $n_i > m$ and the system (2.2.15) is under-determined (we have more unknown coefficients than we have equations). Consequently there might be either none or infinitely many stencils which fulfill the consistency conditions. As discussed in [135], the former case can only occur if the points take certain configurations. For example, the author describes a case where there are as many neighboring points as consistency conditions but they are distributed around the center point in such a way that it is not possible to find a solution for the laplacian stencil. These special point configurations are, however, extremely unlikely in the context of a Lagrangian GFDM where points are moving with continuum velocity and typically do not assume any ordered state. For now we assume that such special cases do not occur due to point movement and the implementation of our point cloud management and refer to the above source for a more in-depth discussion. Before we consider the method we use to calculate suitable stencils among the infinitely many solutions, we want to provide the reader with an example which helps to internalize the above derivations in a less convoluted notation.

Example 2.1 *Let us assume that we want to construct the stencils*

$$\boldsymbol{c}_i^0,\ \boldsymbol{c}_i^x,\ \boldsymbol{c}_i^y,\ \boldsymbol{c}_i^{\nabla^2} \qquad i = 1, \ldots, N \tag{2.2.16}$$

in two space dimensions, i.e. $d = 2$*, and from monomials up to second order, i.e.* $\mathrm{P} = 2$*. As mentioned before, an application of this set of stencils is, for example, the spatial discretization of incompressible Navier-Stokes equations. With these choices, the set of monomials in* (2.2.8) *becomes*

$$\mathcal{M}_i = \{1, x - x_i, y - y_i, (x - x_i)(y - y_i), \frac{1}{2}(x - x_i)^2, \frac{1}{2}(y - y_i)^2\} \tag{2.2.17}$$

where x_i, y_i denote the first and second component of the position vector $\boldsymbol{x}_i$ in 2D. If we introduce the notations $\Delta x_{ij} = x_j - x_i$, $\Delta y_{ij} = y_j - y_i$ for $j \in \mathcal{N}_i$ the matrix columns in (2.2.13) *can simply be written as*

$$\mathrm{M}_i^j = \left(1, \Delta x_{ij}, \Delta y_{ij}, \Delta x_{ij}\Delta y_{ij}, \frac{1}{2}(\Delta x_{ij})^2, \frac{1}{2}(\Delta y_{ij})^2\right)^T \in \mathbb{R}^6 \tag{2.2.18}$$

We recall that the neighborhood definition (2.1.2) *includes the center point for which the above vector reduces to*

$$\mathrm{M}_i^i = (1, 0, 0, 0, 0, 0)^T \in \mathbb{R}^6 \tag{2.2.19}$$

Since we further assumed that $j_1 = i$, the complete matrix takes the form

$$\mathbf{M}_i = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 0 & \Delta x_{ij_2} & \dots & \Delta x_{ij_{n_i}} \\ 0 & \Delta y_{ij_2} & \dots & \Delta y_{ij_{n_i}} \\ 0 & \Delta x_{ij_2}\Delta y_{ij_2} & \dots & \Delta x_{ij_{n_i}}\Delta y_{ij_{n_i}} \\ 0 & \frac{1}{2}(\Delta x_{ij_2})^2 & \dots & \frac{1}{2}(\Delta x_{ij_{n_i}})^2 \\ 0 & \frac{1}{2}(\Delta y_{ij_2})^2 & \dots & \frac{1}{2}(\Delta y_{ij_{n_i}})^2 \end{bmatrix} \tag{2.2.20}$$

The right-hand sides in (2.2.12) *are further given by*

$$\boldsymbol{b}^0 = (1, 0, 0, 0, 0, 0)^T \tag{2.2.21}$$

$$\boldsymbol{b}^x = (0, 1, 0, 0, 0, 0)^T \tag{2.2.22}$$

$$\boldsymbol{b}^y = (0, 0, 1, 0, 0, 0)^T \tag{2.2.23}$$

$$\boldsymbol{b}^{\nabla^2} = (0, 0, 0, 0, 1, 1)^T \tag{2.2.24}$$

If we were to consider points in $\mathbb{R}^2$ that are uniformly distributed along the x- and y-direction and assume that the neighboring points form the classical five-point stencil, it is easy to show that the system (2.2.15) *with matrix* (2.2.20) *and right-hand sides* (2.2.22)-(2.2.24) *leads to the classical central difference stencils (cf. [136]). This is true despite the fact that we have more consistency conditions than points, only because the fourth row in* (2.2.20) *and fourth component of all right-hand sides equals zero in this configuration. It represents one of the extremely unlikely cases which could technically occur in moving point clouds (for more special cases, see [135]).*

Now, let us return to the task of determining suitable stencils from the under-determined systems. As hinted at before, this is typically done in GFDMs by a minimization procedure. In the polynomial approach, the functional which we minimize is given by

$$J_i = \frac{1}{2}\sum_{j \in \mathcal{N}_i}\left(\frac{c_{ij}^*}{w_{ij}}\right)^2 = \frac{1}{2}\left\|\mathbf{W}_i^{-1}\boldsymbol{c}_i^*\right\|^2 \tag{2.2.25}$$

with $\mathbf{W}_i$ denoting the diagonal matrix

$$\mathbf{W}_i = \mathrm{diag}(w_{ij_1}, \dots, w_{ij_{n_i}}) \in \mathbb{R}^{n_i \times n_i} \tag{2.2.26}$$

The components w_{ij} of this matrix represent the results of evaluating a weigthing function $w : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ with respect to the relative distances

$$r_{ij} = \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{h_{ij}} \tag{2.2.27}$$

As was touched upon before, this specific form of the denominator is chosen to ensure that

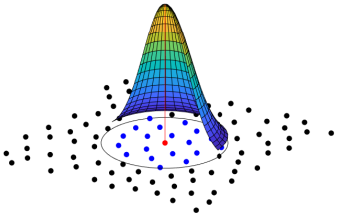


Figure 2.3: Gaussian weighting function centered in a FPM point (red) and plotted within a radius given by the smoothing length.

the distances are symmetric in the sense $r_{ij} = r_{ji}$ and the points are included in each others neighborhood. While different choices for this weighting function are possible in the GFDM [9, 73], we typically use a compactly supported Gaussian (see Figure 2.3), i.e.

$$w(r) = \begin{cases} e^{-\alpha r^2} & r \leq 1 \\ 0 & r > 1 \end{cases} \tag{2.2.28}$$

With the terms in (2.2.25) being clarified, we want to note that the, at first glance, peculiar form of the functional with an inverse of $\mathbf{W}_i$ giving less weight to the center point is a consequence of the weighted minimization

$$\tilde{J}_i = \sum_{j \in \mathcal{N}_i} w_{ij}^2 \epsilon_{ij}^2 \tag{2.2.29}$$

of errors ϵ_{ij} in the equivalent Taylor expansion argument. In one spatial dimension, these errors would be given through

$$\epsilon_{ij} + f(x_j) = f(x_i) + \frac{\partial f}{\partial x}(x_i)(x_j - x_i) + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(x_j - x_i)^2 \tag{2.2.30}$$

For more details on this subject-matter we refer once more to [144]. With these definitions one can prove the following result:

Theorem 2.1 *Consider the minimization problem*

$$\underset{\boldsymbol{c}_i^* \in \mathbb{R}^{n_i}}{\text{minimize}} \quad J_i(\boldsymbol{c}_i^*) = \frac{1}{2}\left\|\mathbf{W}_i^{-1}\boldsymbol{c}_i^*\right\|^2 \tag{2.2.31}$$

$$\text{subject to} \quad \mathbf{M}_i \boldsymbol{c}_i^* = \boldsymbol{b}_i^* \tag{2.2.32}$$

If $\mathbf{M}_i \in \mathbb{R}^{m \times n_i}$ *has full rank, the unique solution to this problem is given by*

$$\boldsymbol{c}_i^* = \mathbf{W}_i^2 \mathbf{M}_i^T \left(\mathbf{M}_i \mathbf{W}_i^2 \mathbf{M}_i^T\right)^{-1} \boldsymbol{b}^* \tag{2.2.33}$$

Proof: For a proof of this statement via Lagrange multipliers we refer the reader to the work by Seibold [135]. □

With this result it is clear that we can first solve the linear system

$$\mathbf{M}_i \mathbf{W}_i^2 \mathbf{M}_i^T \boldsymbol{\lambda}_i = \boldsymbol{b}^* \tag{2.2.34}$$

with the quadratic matrix $\mathbf{M}_i\mathbf{W}_i^2\mathbf{M}_i^T \in \mathbb{R}^{m\times m}$ for the Lagrange multipliers $\boldsymbol{\lambda}_i \in \mathbb{R}^m$, which can then be used in

$$\boldsymbol{c}_i^* = \mathbf{W}_i^2\mathbf{M}_i^T\boldsymbol{\lambda}_i \tag{2.2.35}$$

to obtain the differential operator stencils. It has already been noted in the original work by Liszka and Orkisz [93] that the main difficulty in applying the presented approach is due to singularities or ill-conditioned matrices in linear systems. While the matrix $\mathbf{M}_i\mathbf{W}_i$ in (2.2.34) is typically well conditioned [73], local distortion of point clouds can lead to a degradation of its condition number, which, to make matters even worse, becomes squared in (2.2.34). While this is less of a problem when considering static point clouds, which can be adjusted once to ensure numerical stability, this is a critical point in the Lagrangian approach, since point clouds are constantly evolving. Therefore, it is typically not desirable to directly use this approach. An alternative approach which avoids the calculation of $\mathbf{M}_i\mathbf{W}_i^2\mathbf{M}_i^T$ and solution of (2.2.34) is described in [144] and based on a QR decomposition. A problem which can not be solved by the QR decomposition approach is the case of point clouds which locally become close to lesser dimensional manifolds such as lines in 2D or planes in 3D. At this point we only want to note that these rare cases lead to very small or even zero eigenvalues within $\mathbf{M}_i\mathbf{W}_i^2\mathbf{M}_i^T$ which can, for example, be treated by using a pseudo-inverse calculation [73]. Many other measures are taken within our method in order to ensure stable differential operators for given local distributions of neighbors. It would be far beyond the scope of this thesis to provide a comprehensive overview on all of these and others that have been investigated, which is why we stop our discussion on this subject-matter. In the following section we will, however, still discuss ways to improve the given point clouds instead.

An aspect which we did not address so far is the fact that we can choose different test functions or extend the monomial basis by additional function to impose additional restriction on the stencils [144]. A special example of the latter type, also given in [144], is the addition of delta functions to improve diagonal dominance in the Laplacian stencils on circular neighborhoods. This is especially important when implicitly solving PDEs such as heat or diffusion equations where we want to have a PDE discretization which has the property of (best possible) positivity. We note that the case of such stencils on non-circular neighborhoods has been presented in [135] but is not be utilized in the context of this work, instead we restrict ourselves to the neighborhood definition as presented above. Finally one can also alter the functional that is minimized. An approach which follows this line of thought has been proposed in [153] where the authors include the partial differential equations in a discretized form within the functional and thus the optimization procedure. While this has historically been applied in the FPM and a comparison of this approach with the above-presented method has recently been presented in [146], it does not find application within the context of this work.

2.3 Lagrangian GFDM and point cloud management

In the above section, we already mentioned that due to the Lagrangian nature and therefore constantly changing point clouds, special care needs to be taken in the calculation of differential operators. In the following we want to discuss a few aspects of point cloud management that are important to consider in our Lagrangian GFDM approach.

First, in order to maintain the previously stated number of neighbors, we need to continuously check for holes within the point cloud and fill new points whenever needed. Such holes may, for example, be the result of point movement at inflow boundaries or due to local flow conditions. Whenever a new point is created as a consequence of a hole filling step one can use the function approximation stencil $\boldsymbol{c}_i^0$ to approximate all necessary quantities at the new

point position $\boldsymbol{x}_i$ from values at neighboring points. Obviously, this is not necessary within the filling steps which are performed during the initial point cloud creation, since there the specified initial conditions are applied to newly created points.

Secondly, depending on the flow configuration, clusters with very small point distances can occur, which is undesirable from a performance standpoint and may lead to numerical instabilities in the differential operator calculation. Thus, if the distance between two points becomes smaller than a predefined fraction of the local smoothing length, they are merged, meaning that they are replaced by a single point. Due to the spatial proximity of the old points, the position of the new point is of little importance and one can, for example, place it in the middle between the old point positions. However, as is discussed in more detail in [47, 135], care needs to be taken when choosing the new function values since adopting either of the two point values without any interpolation applied, may lead to an incorrect approximation of discontinuous solutions. Typically this is circumvented by either taking the average of both values or by employing the function approximation stencil at the new point position [40].

We observe that the above measures of maintaining point cloud quality necessitate a treatment of point movement by explicit time integration schemes. A detailed account of different methods is given in [145]. Within this work we have restricted ourselves to the most commonly applied second order movement of the form

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \boldsymbol{u}^n \Delta t + \frac{1}{2} \frac{\boldsymbol{u}^n - \boldsymbol{u}^{n-1}}{\Delta t} \Delta t^2 \tag{2.3.1}$$

where $\boldsymbol{x}^n$, $\boldsymbol{u}^n$ denote the point position and continuum velocity at the discrete time level $t^n = n\Delta t$. Independent of the specific form of point movement, the explicit nature of the overall scheme introduces a time step restriction. Since our numerical stencils depend on points within a radius of size h, the CFL condition when solving the fluid dynamical equations from Chapter 1 locally takes the form

$$\Delta t \leq \mathcal{C} \frac{h_i}{\|\boldsymbol{u}_{\mathrm{f},i}\|_2} \tag{2.3.2}$$

with a constant $\mathcal{C} \in (0, 1)$ which depends on $r_{\min}$ and r_{hole}. In particular, this ensures that points can not leave their current neighborhood within the next time step. From this, we further formulate the global time step restriction

$$\Delta t \leq \mathcal{C} \frac{h_{min}}{\|\boldsymbol{u}_{\mathrm{f},i}\|_{2,max}} \tag{2.3.3}$$

$$\|\boldsymbol{u}_{\mathrm{f}}\|_{2,max} = \max_{i=1,\dots,N} \|\boldsymbol{u}_{\mathrm{f},i}\|_2 \tag{2.3.4}$$

As has been discussed in [136] there is also a criterion which results from the requirement that particle paths do not cross within a single time step. However, condition (2.3.3) is actually the more stringent one and the one which is used to determine time steps when an adaptive time stepping is employed.

To circumvent the time step restrictions which follow from the Lagrangian nature of our approach, one can also keep the point cloud static, albeit at the cost of having to resolve the convective terms in Eulerian formulations of the underlying differential equations. For an in-depth discussion on this subject-matter and solution approaches within the finite pointset method we refer the interested reader to the work by Seifarth [136].

Aside from considerations of point cloud quality and numerical stability, we also need to consider computational cost. Due to both, the point movement and management, the

neighbor relations described in the first section of this chapter are continuously changing and need to be updated. This process is one of the most time consuming parts of the meshfree GFDM approach. In the most naive implementation, one could recompute the lists of neighbors in every time step by checking the distance between every pair of points against the smoothing length. The implication of this procedure would be a growth of computational cost per time cycle with $\mathcal{O}(N^2)$, which is clearly unacceptable. Instead one can construct a data structure which represents a regular grid of elements in $\mathbb{R}^d$ (squares in 2D or cubes in 3D) with side length h so that the computational domain is covered by these "buckets" and neighbors of every point can be found in either the buckets the point is located in or its directly adjacent ones. As is described in [40], by maintaining lists that map between the bucket indices and point indices, the computational cost of neighbor search can be reduced to $\mathcal{O}(N \log(N))$. However, it is discussed in [94], that this approach becomes increasingly inefficient with stronger variations in smoothing length. Therefore, we employ an approach based on a binary search tree structure which does not suffer from this deficiency [143]. We further want to note that our implementation of the FPM, i.e. the software MESHFREE, is able to utilize the capabilities of parallel computing on distributed memory via the Message Passing Interface (MPI) standard. This allows for the solution of continuum mechanical problems on point clouds with millions of points [73].

Chapter 3

Multiphase flow

In the last two chapters we have introduced the equations which govern the motion of fluid phases and the framework in which we intend to solve our original problem, the coupling of fluid and solid phases. It is within this chapter that we want to discuss solution approaches and provide an introduction to the theoretical basis for our numerical investigations.

The chapter is structured as follows: We first give a short overview on different models which are used for the numerical simulations of coupled fluid-solid flows in the literature and provide the initial motivation for the approach we use in this thesis. We then introduce the theoretical framework of this method and provide more details on its different realizations in the literature and the coupling terms that might be involved.

3.1 Models for fluid-solid coupling

The coupled numerical simulation of fluids with immersed solid particles is an important tool for a wide range of industrial applications. Historically, a big driver for the development of such methods can be found in chemical engineering applications (see for example [173] and references therein). A broad overview of different coupling approaches in the context of gas-solid fluidized beds is presented by Hoef et al. [64], more generally for dense particulate reaction systems by Zhong et al. [173], and for the coupling of continuous and dispersed phases by Loth [95] and van Wachem, Almstedt [161]. For text books which provide more extensive discussions on multiphase models in general, or fluid-solid flows in particular we refer the reader to [26, 55, 108, 118]. In the following we typically refer to a single entity of solid material as *particle* in order to provide clear distinction from the numerical *points* in the FPM.

There are several factors which influence the choice of a fitting approach to this coupling. If we consider a single particle that moves at a low velocity in a comparatively large volume of fluid and with no proximity to other particles, it is very likely that this particle does not influence the fluid flow in any significant way. In such cases a so-called *one-way coupling* is applicable, meaning that forces exerted by the fluid phase onto individual particles are calculated but the solid particles are essentially invisible to the fluid phase. However, as particle size increases or they become clustered, there will be a significant influence by the bulk of solid entities onto the fluid phase. In such cases it is important to consider a *two-way coupling* where also the fluid flow is, in some way, influenced by the dispersed phase. The specific form of this influence depends on the chosen approach and will be clarified in the upcoming sections. We also note that in very *dilute* suspensions, i.e. a mixture of a carrier fluid and immersed solid entities where the average distance between particles is relatively large, it is likely sufficient to consider particles which do not interact with each other and whose movement is solely determined by body forces and the interaction with the fluid phase. However, with this strong simplification it is not possible to represent cases of accumulations which confine the flow domain or breakage induced by collisions. Thus, within this work, we devise a scheme for the most general case involving interacting particles, with the

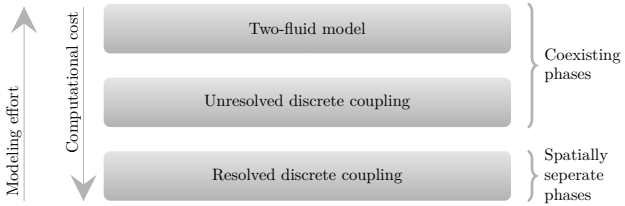


Figure 3.1: Overview of different approaches for the simulation of coupled fluid-solid flows.

reduction to non-interacting ones being obvious. We note that more detailed classifications of flow conditions and guidelines for assessing the suitability of different coupling degrees are presented, for example, in [27, 108]. These are mostly based on the relation of characteristic time scales of the system to the so-called *particle relaxation time*. This quantity describes the relaxation of particle velocity towards fluid velocity and is properly introduced in Section 4.6. We further point out that a bidirectional coupling of fluid and solid phase in combination with interacting particles is denoted as four-way coupling by the authors in [27, 108]. While this is a convention that can be found in some part of the literature on multiphase flows, we stick to the above differentiation between cases where momentum exchange is unidirectional or bidirectional and denote them as one-way and two-way coupled.

To provide an overview of coupling methods we roughly follow the classification in [64] and consider them with decreasing computational cost but increasing amount of modeling involved (see Figure 3.1). A reason for this specific choice of naming convention for the coupling approaches will be given at a later point. Note that we omit models which consider scales where the continuum theory developed in Chapter 1 looses its validity, such as the molecular dynamics model mentioned in [64]. Instead, we first want to consider what we refer to as *resolved discrete coupling* in the following.

Resolved discrete coupling

As described, for example, in [118], particles in the solid phase experience a fluid-solid interaction force $\boldsymbol{F}_{\mathrm{fs}}$ due to interior stresses of the surrounding fluid phase, which can be obtained by integrating the fluid stress tensor $\boldsymbol{\sigma}_{\mathrm{f}}$ over the surface of a particle, i.e.

$$\boldsymbol{F}_{\mathrm{fs}} = \int_{S_{\mathrm{p}}} \boldsymbol{n}_{\mathrm{s}} \cdot \boldsymbol{\sigma}_{\mathrm{f}} \, \mathrm{d}S \tag{3.1.1}$$

where $\boldsymbol{n}_{\mathrm{s}}$ denotes the outward pointing unit normal of the particle surface S_{p}. Assuming we are able to compute the fluid flow in a coupled numerical simulation so that all fluctuations close to particle surfaces are resolved, the evaluation of the above integral suffices to fully represent the forces experienced by the particle, without any modeling involved. On the other hand, the influence of particles onto the fluid is ensured by considering them as moving boundaries to the fluid flow domain. Thus, in a resolved discrete coupling scheme one considers each solid particle as a single discrete entity and solve a single-phase flow problem in the fluid phase with a numerical discretization that is able to resolve each particle surface

sufficiently well. The translational movement of each discrete particle is then computed by the integration of its equations of motion

$$\frac{\mathrm{d}\boldsymbol{x}_\mathrm{s}}{\mathrm{d}t} = \boldsymbol{v}_\mathrm{s} \tag{3.1.2}$$

$$m_\mathrm{s}\frac{\mathrm{d}\boldsymbol{v}_\mathrm{s}}{\mathrm{d}t} = \boldsymbol{F}_\mathrm{fs} + \boldsymbol{F}_\mathrm{b} + \boldsymbol{F}_\mathrm{c} \tag{3.1.3}$$

where $\boldsymbol{x}_\mathrm{s}$, $\boldsymbol{v}_\mathrm{s}$, m_s, $\boldsymbol{F}_\mathrm{b}$, $\boldsymbol{F}_\mathrm{c}$ denote the particle position, velocity, mass, other body forces and forces due to collisions with other particles or boundaries, respectively. Typically when such a discrete representation of the solid phase is considered, one does not resolve internal stresses and deformations due to collisions. Instead, simplified models, partially derived from elasticity theory, are employed to reduce computational cost. One of the most common methods of this type is the so-called *discrete element method (DEM)*, in which the discretely represented solid entities following (3.1.2), (3.1.3) are allowed to slightly overlap and forces which act to recover the non-overlapping state are calculated from a measure of overlap size as well as relative velocities. More details on this methodology are omitted here and will be provided in Chapter 4, since they are not necessary for the current discussion. The combination of a continuous phase solver for the fluid phase and the discrete element method for the solid phase is often referred to as *CFD-DEM*.

Even when particles are not close to each other (dilute suspension), it is evident that, since the numerical resolution in the fluid phase is dictated by particle sizes, this approach quickly becomes infeasible when they are very small compared to the flow domain. This problem is further aggravated in dense configurations where, for example, liquid bridges [26] causing adhesive forces might need to be resolved in small gaps between the particles. One might argue that such small scale effects can be captured with some additional modeling effort instead of enforcing an increase in fluid phase resolution, however, a strict dependency on particle size remains. From these considerations, the necessity for a method that relaxes the requirements on spatial discretization in the continuous phase while keeping the discrete nature in the solid phase becomes apparent. One class of methods which approach this problem are those based on the *unresolved discrete coupling* introduced in the following.

Unresolved discrete coupling

The general idea of an unresolved discrete coupling is that individual particle surfaces are no longer resolved and the fluid flow is calculated on a scale above the particle dimensions. Instead of being treated as moving boundaries to the continuous phase, the dispersed phase is treated as a collection of point sources and sinks of momentum. In particular, the two phases are not spatially separated any more, i.e. the particles do not exclude volume in the computational domain from the fluid. To see which effect this has on the form of (3.1.1), we decompose the fluid stress $\boldsymbol{\sigma}_\mathrm{f}$ into a part $\langle\boldsymbol{\sigma}_\mathrm{f}\rangle$ which can be resolved by the coarse spatial discretization in the fluid phase and a part $\boldsymbol{\sigma}'_\mathrm{f}$ which subsumes small scale fluctuations, i.e.

$$\boldsymbol{\sigma}_\mathrm{f} = \langle\boldsymbol{\sigma}_\mathrm{f}\rangle + \boldsymbol{\sigma}'_\mathrm{f} \tag{3.1.4}$$

Then we can write

$$\boldsymbol{F}_\mathrm{fs} = \int_{S_\mathrm{p}} \boldsymbol{n}_\mathrm{s} \cdot \langle\boldsymbol{\sigma}_\mathrm{f}\rangle \,\mathrm{d}S + \int_{S_\mathrm{p}} \boldsymbol{n}_\mathrm{s} \cdot \boldsymbol{\sigma}'_\mathrm{f} \,\mathrm{d}S \tag{3.1.5}$$

Using the divergence theorem and the fact that $\langle\boldsymbol{\sigma}_\mathrm{f}\rangle$ can be assumed to be approximately constant within a single particle, we arrive at

$$\boldsymbol{F}_\mathrm{fs} \approx (\boldsymbol{\nabla} \cdot \langle\boldsymbol{\sigma}_\mathrm{f}\rangle)V_\mathrm{s} + \int_{S_\mathrm{p}} \boldsymbol{n}_\mathrm{s} \cdot \boldsymbol{\sigma}'_\mathrm{f} \,\mathrm{d}S \tag{3.1.6}$$

where V_s denotes the particle volume. Therefore, if we are solving our fluid phase on a scale above the individual particle level, the obtained stresses can be directly used to evaluate the first term. For the second term it is necessary to find an expression that links the forces due to small scale fluctuations to fields available on the fluid scale. This is typically done by providing individual formulas for each of the physical phenomena which are products of these fluctuations, such as drag, lift or virtual mass forces. These formulas are obtained as empirical correlations either from experiments or resolved simulations and heavily depend on the considered problem. In particular, while the resolved coupling doesn't require special treatment of different particle shapes or configurations of multiple particles, in an unresolved coupling a strong deviation in either of these factors requires a modification of the correlation. More details on these expressions are provided in Section 3.4.

With suitable expressions for the forces onto particles at hand, it is further necessary to provide momentum source terms in the fluid phase which obey Newton's third law. This calculation is typically done by some kind of averaging procedure which transfers the quantities attached to single particles onto the fluid phase discretization. The realization of this step is a major differentiating factor between different CFD-DEM schemes found in literature. More details on this procedure, often also referred to as coarse graining, are given in Section 4.2.

While the unresolved approach significantly reduces the computational expense in the continuous phase, the solid phase is still represented discretely. Thus, the cost associated with the neighbor search for resolving collisions remains. Furthermore, the computational effort for both, interpolations onto each solid particle and the aforementioned averaging, scale with the granular system size. Especially the latter procedure comes at a significant cost and takes up a major part of the simulation time in an unresolved coupling scheme. To alleviate this problem, methods have been proposed which subsume many of the physical particles into representative numerical entities such as the MP-PIC (see [173] and references therein). But even with such approaches, industry-scale problems can include numbers of particles large enough such that tracking every single solid entity still comes at a prohibitively large computational cost.

However, in some of these applications it might be sufficient to only consider macroscopic averages of the individual particle properties. In such cases a direct computation of these quantities can drastically reduce the numerical complexity in the solid phase. A class of methods for coupled fluid-solid flows which build upon this idea are those using a *two-fluid model* (TFM).

Two-fluid model

As the name already suggests, the macroscopic solid phase behavior in two-fluid models is typically described by a similar set of equations as the fluid phase. Thus, the viewpoint of individual particles is entirely dropped and they are treated as a continuous phase with a stress tensor and viscosity which are formulated based on the bulk motion of particles. There is a variety of different theories to perform the necessary transition from individual particle behavior to macroscopic behavior in a bulk of particles, but the most commonly used one is based on the kinetic theory of granular flow [55, 161], which uses concepts from kinetic gas theory. Typically the general form of momentum exchange terms in two-fluid models is the same as in the unresolved discrete coupling, the only difference is the fact that they are no longer evaluated for individual particles and then averaged. Instead, they are calculated directly based on the macroscopic flow quantities of both phases.

Using models of this type enables the simulation of large-scale industrial problems due to the drastically reduced computational cost within the solid phase. Furthermore, in case

the particulate phase is modeled with similar equations as the fluid phase, one can also use similar numerical schemes for both phases.

However, the efficiency of this approach does come at the cost of increased modeling effort. Rather similar to the transition from unresolved to resolved discrete coupling, one needs to reflect small-scale phenomena within the solid bulk that are no longer resolved. These can include the formation of patterns such as particle bridges, which in turn may lead to clogging within the flow geometry or the segregation of particles in polydisperse suspensions. Additionally, as noted by Crowe et al. [26], the TFM only makes sense in dense particulate systems since only then will information (temperature, velocity) travel in all directions, i.e. diffuse, also in the discrete model. For dilute systems the information follows the motion of individual particles instead.

This concludes our brief overview on different coupling approaches and we now discuss our choice of method in the context of the present work. There are specific applications which can lead to models with even more different combinations of scales, like the discrete bubble model mentioned in [64], which is specific to gas-solid fluidization. However, we did want to restrict our attention towards more general models that are applicable in a wider range of problems and exclude specialized approaches. We also note that, due to the prevalence of mesh-based methods in literature, it is common practice to denote phases with a continuous description as "Eulerian" phase, while those with a discrete one are referred to as "Lagrangian" phase. In this naming convention one often finds a "Eulerian-Eulerian coupling" to be synonymous with a two-fluid model, while a "Eulerian-Lagrangian coupling" can either refer to a resolved or unresolved discrete coupling (cf. [173]). Clearly, this type of naming convention is not suitable for our numerical framework, which is why we resorted to the one depicted in Figure 3.1 and used throughout the current discussion.

Discussion of models in FPM context

When we consider the above approaches within the Lagrangian GFDM framework of Chapter 2, it is apparent that the integration of a discrete particle method is much more natural than it is in a mesh-based Eulerian framework. For the detection of collisions we can make use of the neighbor search algorithms which are necessary for the construction of differential operators. These algorithms form the major part of computational expense in DEM methods and the corresponding data structures and parallelizations are already available and have the capacity to handle millions of points in our framework. This fact is one of the main reasons why we choose to not pursue a two-fluid approach in favor of a scheme with discrete solid phase.

Among these schemes, let us first consider the resolved discrete particle model. In this approach, one could make use of the fact that the Lagrangian FPM is particularly suitable for flows with moving boundaries, mainly because there is no need for repeated re-meshing of the domain. This makes the approach rather flexible in terms of particle shape and would certainly be a useful tool for highly resolved simulations. Their results can then be utilized to obtain expressions which model small-scale phenomena for an unresolved coupling. However, for flow problems beyond academic scales it is evident that this approach is not applicable due to its computational expense and the numerical problems induced in the fluid phase by dense configurations of particles.

While we retain the details in solid particle movement and can still make use of the unified Lagrangian framework, an unresolved coupling allows us to consider problem scales which are much closer to industrial applications. This is why we choose to pursue the unresolved coupling approach within the context of this work.

We note that a coupling with non-interacting droplets of fluid has already been realized in the considered framework by Drumm et al. [40]. However, the approach within this work does differ from their scheme in many aspects that are discussed at a later point in Chapter 4. Most notably, we consider how averaging approaches from mesh-based CFD-DEM can be translated to our setting, despite the absence of static computational cells and the corresponding natural notion of averaging volumes. In the course of this discussion we also highlight the difference in our approach compared to mesh free methods which are based on mass particles, such as SPH.

In the remainder of this chapter we lay the theoretical foundations for the chosen coupling, before we turn towards their numerical treatment in Chapter 4. First, we introduce the averaging formalism that leads to a set of generalized Navier-Stokes equations in the fluid phase and dictates how the momentum exchange terms are defined. With this knowledge we then consider different versions of unresolved coupling and discuss the necessary empirical expressions for coupling forces that result from phenomena below the averaging scale.

3.2 Volume averaged Navier-Stokes equations

The foundations for methods like unresolved CFD-DEM and the two-fluid model have been laid in the 1960s. During that time, several authors independently derived the spatial averaging theorem which is central to the derivation of those partial differential equations underlying the above coupling approaches [4, 101, 138, 164, 166]. Although their results were very similar, the notion of an averaged quantity is different in each of the original contributions. As is detailed in [120] and more recently in [32], the most general of these definitions is based on the approach by Marle [101], which makes use of distribution theory. On the other hand, their formalism is rarely referenced in the context of unresolved CFD-DEM, while the weighted approach due to Anderson and Jackson [4] and unweighted ones due to Whitaker [164, 165, 166] and Slattery [138] are much more common. There is also a variety of other approaches that lead to similar equations, such as the ensemble averaging approach by Zhang and Prosperetti [170, 171]. Although we only employ Anderson and Jacksons method to derive the equations we aim to solve numerically, it is worthwhile to consider a framework that unifies some of the formalisms and helps to put the approach in this work and in CFD-DEM literature into context. To achieve this, let $\psi : \mathbb{R}^d \to \mathbb{R}$ be a quantity which is present in both phases so that

$$\psi = \psi_\mathrm{f}\chi_\mathrm{f} + \psi_\mathrm{s}\chi_\mathrm{s} = \begin{cases} \psi_\mathrm{f} & \text{in the fluid phase} \\ \psi_\mathrm{s} & \text{in the solid phase} \end{cases} \tag{3.2.1}$$

where χ_f, χ_s denote the indicator functions on the fluid and solid volume, while ψ_f, ψ_s are the fields within a single phase. The generalized averaging formalism, described in [119, 120], is based on the convolution

$$(w * \psi)(\boldsymbol{x}) = \int_{\mathbb{R}^d} w(\boldsymbol{x} - \boldsymbol{y})\psi(\boldsymbol{y}) \,\mathrm{d}V_y \tag{3.2.2}$$

with a kernel w that is compactly supported and normalized in the sense that

$$\int_{\mathbb{R}^d} w \,\mathrm{d}V = 1 \tag{3.2.3}$$

This allows us to define the *superficial fluid phase average*

$$\langle \psi_\mathrm{f} \rangle = w * (\chi_\mathrm{f}\psi) \tag{3.2.4}$$

and *intrinsic fluid phase average*

$$\langle \psi_\mathrm{f} \rangle^\mathrm{f} = \frac{w * (\chi_\mathrm{f} \psi)}{w * \chi_\mathrm{f}} \tag{3.2.5}$$

in a way that is, through the choice of appropriate weighting kernels, consistent with the specialized averaging procedures discussed in the introduction to this section. One of the differences between the two definitions is the fact that quantities which are constant within a phase are exactly reproduced by the corresponding intrinsic averages while the superficial average can differ. Within this work we are mainly interested in intrinsic averages, which we may analogously defined for the solid phase, i.e.

$$\langle \psi_\mathrm{s} \rangle^\mathrm{s} = \frac{w * (\chi_\mathrm{s} \psi)}{w * \chi_\mathrm{s}} \tag{3.2.6}$$

The denominators in (3.2.5) and (3.2.6) also play a key role in the equations we want to derive. The term

$$\varepsilon_\mathrm{f} = w * \chi_\mathrm{f} \tag{3.2.7}$$

denotes the *fluid volume fraction*, also called *porosity* or *voidage*, while

$$\varepsilon_\mathrm{s} = w * \chi_\mathrm{s} \tag{3.2.8}$$

is the *solid volume fraction*. As their name suggests, these quantities describe how much of a certain (averaging) volume is occupied by the respective phase. Due to the kernel being normalized, the volume fractions further fulfill

$$\varepsilon_\mathrm{f}(\boldsymbol{x}) + \varepsilon_\mathrm{s}(\boldsymbol{x}) = 1 \quad \forall\, \boldsymbol{x} \in \mathbb{R}^d \tag{3.2.9}$$

which represents the intuitive requirement that, in a two-phase flow, adding up the fluid and solid volume should provide the whole volume on which the fractions were calculated. To better visualize this and also to see how the different averaging approaches can be accommodated into the convolution formalism, let us first consider the classical *unweighted averaging*. This definition due to Slattery [138] can also be found in a variety of textbooks on multiphase flow [26, 118] and is given by

$$\langle \psi_\mathrm{f} \rangle^\mathrm{f}(\boldsymbol{x}) = \frac{1}{|\mathcal{V}_\mathrm{f}|} \int\limits_{\mathcal{V}_\mathrm{f}(\boldsymbol{x})} \psi_\mathrm{f} \,\mathrm{d}V \tag{3.2.10}$$

with the fluid volume fraction being defined as

$$\varepsilon_\mathrm{f} = |\mathcal{V}_\mathrm{f}| / |\mathcal{V}| \tag{3.2.11}$$

Herein, an averaging volume $\mathcal{V} \subset \mathbb{R}^d$ of fixed size and shape is moved with the point of evaluation so that $\mathcal{V}(\boldsymbol{x})$ is centered in $\boldsymbol{x}$ and $\mathcal{V}_\mathrm{f}(\boldsymbol{x})$ denotes the part of $\mathcal{V}(\boldsymbol{x})$ covered by the fluid phase (see Figure 3.2). Note that we use the calligraphic $\mathcal{V}$ for subsets of $\mathbb{R}^d$ to distinguish it from geometric volumes V and denote the volume of a set as

$$|\mathcal{V}| = \int_\mathcal{V} 1 \,\mathrm{d}V \tag{3.2.12}$$

We observe that the average definition (3.2.10) is recovered from the generalized formalism (3.2.5) by choosing the piecewise constant kernel

$$w_\mathcal{V}(\boldsymbol{r}) = \frac{1}{|\mathcal{V}|} \begin{cases} 1 & \text{if } \boldsymbol{r} \in \mathcal{V} \\ 0 & \text{if } \boldsymbol{r} \in \mathbb{R}^d \setminus \mathcal{V} \end{cases} \tag{3.2.13}$$

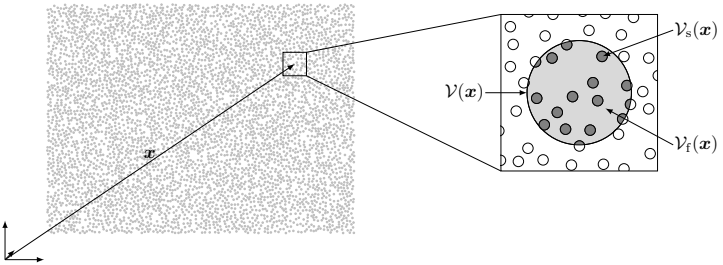


Figure 3.2: Visualization of a spherical moving averaging volume in the unweighted formalism. Light shading in the magnified rectangular region denotes the volume inside $\mathcal{V}(\boldsymbol{x})$ that is covered by fluid, i.e. $\mathcal{V}_\mathrm{f}(\boldsymbol{x})$, while dark shaded gray visualizes the solid phase counterpart $\mathcal{V}_\mathrm{s}(\boldsymbol{x})$.

often referred to as boxcar or top-hat function. For certain problems other non-smooth kernels might also be advantageous, as discussed in [32].

On the other hand, the weighted averaging approach by Anderson and Jackson [4] results from choosing a radially symmetric and smooth kernel given by

$$w(\boldsymbol{r}) = \tilde{w}(\|\boldsymbol{r}\|) \tag{3.2.14}$$

with $\tilde{w} \in \mathcal{C}^\infty(\mathbb{R}_+, \mathbb{R}_+)$. Possible choices for $\tilde{w}$ include be the Gaussian

$$\tilde{w}(r) = C_{b,d} \exp\left(-\frac{r^2}{b^2}\right) \tag{3.2.15}$$

or an element of a standard Dirac sequence

$$\tilde{w}(r) = \varepsilon^{-d} \varphi\left(\frac{r}{\varepsilon}\right) \tag{3.2.16}$$

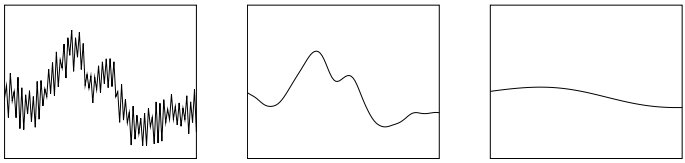
based on a positive symmetric mollifier such as

$$\varphi(r) = \begin{cases} C_d \exp\left(-\dfrac{1}{1-r^2}\right) & \text{if } r < 1 \\ 0 & \text{if } r \geq 1 \end{cases} \tag{3.2.17}$$

In both types of weighting function, $C_{b,d}$ and C_d denote constants that ensure normalization of the kernels on $\mathbb{R}^d$. While the averaging volume for a Gaussian kernel is formally unbounded, the second choice represents a compactly supported function which again leads to a bounded averaging volume.

We recall that we have already encountered the application of averaging to bridge scales when deriving the conservation equations for a continuous phase. There we touched upon the fact that a sensible definition of averages is not always possible. Luckily for multi-phase flows it has been shown that there is an enormous amount of applications for which such a definition can be made in a meaningful way. The common assumption underlying derivations based on the above averages is the existence of a **separation of scales**:

> Assumption: If L_p is the characteristic particle size (radius R_p for spherical particles) and L_c is the smallest length scales at which relevant changes in the continuous phase behavior occur, we assume that $L_c \gg L_p$ holds.



(a) Quantity without averaging **(b)** $L_p < L_w < L_c$ **(c)** $L_c < L_w$

Figure 3.3: Effect of weighted averaging for different kernel length scales L_w, when applied to a fluid quantity with small scale fluctuations. The particle length scale is denoted by L_p while the smallest scale of interest for the continuous phase is denoted by L_c.

Let us define a characteristic length L_w, which represents the range at which the kernel values have a relevant magnitude. For an unweighted averaging this is given by the size of the moving averaging volume, while it is a multiple of the kernel bandwidth in a weighted averaging. Under the above assumption we can infer, that there is a $L_w \in (L_p, L_c)$ so that small scale fluctuations are smoothed out but the macroscopic variations of interest are kept. This idea is conceptually depicted in (see Figure 3.3). In the following we also assume that this averaging enables a splitting of point properties into averages and fluctuating parts

$$\psi = \langle \psi \rangle + \psi' \tag{3.2.18}$$

with

$$\langle \psi' \rangle = 0 \quad \Leftrightarrow \quad \langle \langle \psi \rangle \rangle = \langle \psi \rangle \tag{3.2.19}$$

In particular we make use of the identity

$$\langle \langle \psi_\mathrm{f} \rangle^\mathrm{f} \rangle^\mathrm{s} = \langle \psi_\mathrm{f} \rangle^\mathrm{f} \tag{3.2.20}$$

where we note that the double averaging on the left is possible due to the fluid intrinsic average $\langle \psi \rangle^\mathrm{f}$ being defined on the whole domain, including the solid phase volume, even if ψ is a continuous phase quantity. More in-depth discussions surrounding the separation of scales assumption and derived identities can be found in many of the cited textbooks on multiphase problems and for example in [33]. Here, we won't go into more detail, since this would quickly go far beyond the scope of this thesis and obscure the main target of this section, the derivation of conservation laws in terms of averaged quantities. We emphasize that this derivation is not a novel or original result, but is detailed here for the sake of clarifying the notations and theoretical background of the coupling considered within this work.

For this derivation we closely follow the original work by Anderson and Jackson [4]. As in their work, we replace the integral domain $\mathbb{R}^d$ in (3.2.2) by $\mathcal{V}_\infty \subseteq \mathbb{R}^d$ and formulate the normalization condition as

$$\int_{\mathcal{V}_\infty} w(\boldsymbol{x} - \boldsymbol{y}) \, \mathrm{d}V_y = 1 \tag{3.2.21}$$

We further restrict the possible choice of $\boldsymbol{x}$ to a bounded subset $\mathcal{V} \subset \mathcal{V}_\infty$ which fulfills

$$w(\boldsymbol{x} - \boldsymbol{y}) \approx 0, \quad \forall \, \boldsymbol{x} \in \mathcal{V}, \; \boldsymbol{y} \in \partial \mathcal{V}_\infty \tag{3.2.22}$$

For a compactly supported kernel this condition means that its support is fully inside $\mathcal{V}_\infty$, when centered in $\boldsymbol{x} \in \mathcal{V}$. Since the kernel used in [4] is the Gaussian with unbounded support, the above restriction is formulated a bit weaker, i.e. the kernel is approximately zero at the

domain boundary and the normalization is thus preserved only approximately. This also motivates the subscript notation in $\mathcal{V}_\infty$, since formally the surface of the considered flow domain needs to be infinitely far away from the evaluation point $\boldsymbol{x}$ for the kernel to vanish. Nevertheless, we retain this restriction in the following derivations without putting too much emphasis on the slightly more complex formalism that would be necessary to incorporate averaging close to boundaries. In fact, while the derivations by Anderson and Jackson [4, 72], which also omit this problem, are frequently cited in the literature on unresolved CFD-DEM [39, 62, 75, 161, 175, 177], little attention is devoted to this omission. As is also discussed in [104], it is therefore important to realize that derivations of the multiphase equations are often presented with lack of mathematical rigor in some aspects and understood as a modeling tool. However, the amount of literature where this model is successfully applied to a wide range of applications, illustrates that these simplifications do not give rise to errors that would impede its usefulness. Thus, we follow the above line of thought and consider the fluid and solid volume with respect to the flow domain $\mathcal{V}_\infty$, i.e.

$$\mathcal{V}_\infty = \mathcal{V}_\mathrm{f} \dot{\cup} \mathcal{V}_\mathrm{s} \tag{3.2.23}$$

in the following. Consequently, the normalization condition (3.2.21), together with the assumptions on $\mathcal{V}$, ensures

$$\varepsilon_\mathrm{f}(\boldsymbol{x}) + \varepsilon_\mathrm{s}(\boldsymbol{x}) = 1 \quad \forall\, \boldsymbol{x} \in \mathcal{V} \tag{3.2.24}$$

3.2.1 The averaging theorem

With all assumptions and the averaging formalism being clarified, we now turn our attention towards the derivation of conservation laws formulated in terms of averaged quantities. We do this by multiplying the point-wise mass and momentum conservation equations of the fluid phase with the kernel, followed by an integration over $\mathcal{V}_\mathrm{f}$. This results in integrals over temporal and spatial derivatives of the point-wise quantities, which need to be reformulated in terms of derivatives of the averaged fields. This subsection is devoted to the derivation of two important identities which enable this reformulation.

For the relation which applies to temporal derivatives, we want to reformulate the convolution

$$w * \left(\chi_\mathrm{f} \frac{\partial \psi_\mathrm{f}}{\partial t} \right) = \varepsilon_\mathrm{f} \langle \frac{\partial \psi_\mathrm{f}}{\partial t} \rangle^\mathrm{f} \tag{3.2.25}$$

To achieve this, we apply the Reynolds transport theorem in the form

$$\begin{aligned} \frac{\partial}{\partial t} \int\limits_{\mathcal{V}_\mathrm{f}(t)} \psi_\mathrm{f}(\boldsymbol{y},t) w(\boldsymbol{x}-\boldsymbol{y}) \,\mathrm{d}V_y = & \int\limits_{\mathcal{V}_\mathrm{f}(t)} \frac{\partial \psi_\mathrm{f}}{\partial t}(\boldsymbol{y},t) w(\boldsymbol{x}-\boldsymbol{y}) \,\mathrm{d}V_y \\ & + \int\limits_{\partial\mathcal{V}_\mathrm{f}(t)} \psi_\mathrm{f}(\boldsymbol{y},t) w(\boldsymbol{x}-\boldsymbol{y}) (\boldsymbol{v}_b \cdot \boldsymbol{n}_\mathrm{f}) \,\mathrm{d}S_y \end{aligned} \tag{3.2.26}$$

with $\boldsymbol{v}_b \cdot \boldsymbol{n}_\mathrm{f}$ representing the dot product of the surface element velocity and the outer unit normal to $\partial\mathcal{V}_\mathrm{f}(t)$. We recognize the first integral on the right-hand side as the averaged temporal derivative such that we can write

$$\frac{\partial}{\partial t} \Big(\varepsilon_\mathrm{f}(\boldsymbol{x},t) \langle \psi_\mathrm{f} \rangle^\mathrm{f}(\boldsymbol{x},t) \Big) = \varepsilon_\mathrm{f}(\boldsymbol{x},t) \langle \frac{\partial \psi_\mathrm{f}}{\partial t} \rangle^\mathrm{f}(\boldsymbol{x},t) + \int\limits_{\partial\mathcal{V}_\mathrm{f}(t)} \psi_\mathrm{f}(\boldsymbol{y},t) w(\boldsymbol{x}-\boldsymbol{y}) (\boldsymbol{v}_b \cdot \boldsymbol{n}_\mathrm{f}) \,\mathrm{d}S_y \tag{3.2.27}$$

So far this procedure is identical in all approaches based on the above convolution formalism. In particular the same derivation can be found in literature based on the approaches by

Slattery [138] or Whitaker [164, 166]. To reformulate the second term, we split the total surface of the fluid volume, from now on denoted by $S_\mathrm{f}(t) = \partial\mathcal{V}_\mathrm{f}(t)$, into the interface to solid particles $S_{\mathrm{f,s}}(t)$ and the remaining surface $S_{\mathrm{f},\infty}$ defined by the averaging domain $\mathcal{V}_\infty$. Note that, when we split integrals over $S_{\mathrm{f,s}}(t)$ into parts corresponding to single particles $S_{\mathrm{p},i}(t)$, we usually replace the normals $\boldsymbol{n}_\mathrm{f}$ by those pointing outwards from the particles, i.e. $\boldsymbol{n}_\mathrm{s}$. The surface integral then reads

$$\begin{aligned}&\int\limits_{\partial\mathcal{V}_\mathrm{f}(t)} \psi_\mathrm{f}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{v}_b\cdot\boldsymbol{n}_\mathrm{f})\,\mathrm{d}S_y\\ &= \int\limits_{S_{\mathrm{f},\infty}} \psi_\mathrm{f}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{v}_b\cdot\boldsymbol{n}_\mathrm{f})\,\mathrm{d}S_y - \sum_{i=1}^{N_\mathrm{s}} \int\limits_{S_{\mathrm{p},i}(t)} \psi_\mathrm{f}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{v}_b\cdot\boldsymbol{n}_\mathrm{s})\,\mathrm{d}S_y\end{aligned} \tag{3.2.28}$$

For the far away fluid surface $S_{\mathrm{f},\infty}$ we assume that it is fixed, i.e. $\boldsymbol{v}_b = 0$, while at the particle surfaces $\boldsymbol{v}_b$ coincides with the fluid velocity. If we restrict ourselves to an impermeable particulate phase with constant size (see [14] for a more general formulation with time-varying size), the velocity of fluid and particle along the surface normals have to be identical. With these simplifications we obtain the first important relation

$$\varepsilon_\mathrm{f}\langle\frac{\partial\psi_\mathrm{f}}{\partial t}\rangle^\mathrm{f} = \frac{\partial}{\partial t}\Big(\varepsilon_\mathrm{f}\langle\psi_\mathrm{f}\rangle^\mathrm{f}\Big) + \sum_{i=1}^{N_\mathrm{s}} \int\limits_{S_{\mathrm{p},i}(t)} \psi_\mathrm{f}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{v}_{\mathrm{s},i}\cdot\boldsymbol{n}_\mathrm{s})\,\mathrm{d}S_y \tag{3.2.29}$$

where we omitted the dependence on space and time outside integrals for better readability. Next we want to consider the relation for spatial derivatives, which is referred to as averaging theorem in [120] and was derived in the generalized formalism by the authors. Here we restrict ourselves to the Anderson and Jackson formalism and consider the term

$$\int\limits_{\mathcal{V}_\mathrm{f}(t)} (\boldsymbol{\nabla}\cdot\boldsymbol{a}_\mathrm{f})(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y = \varepsilon_\mathrm{f}(\boldsymbol{x},t)\langle\boldsymbol{\nabla}\cdot\boldsymbol{a}_\mathrm{f}\rangle^\mathrm{f}(\boldsymbol{x},t) \tag{3.2.30}$$

for a smooth weighting function w and vector valued property of the fluid $\boldsymbol{a}_\mathrm{f}(\cdot,t) \in \mathcal{C}^1\Big(\mathbb{R}^d,\mathbb{R}^d\Big)$. In order to rewrite this term we observe that

$$\begin{aligned}\boldsymbol{\nabla}\cdot\Big(\varepsilon_\mathrm{f}(\boldsymbol{x},t)\langle\boldsymbol{a}_\mathrm{f}\rangle^\mathrm{f}(\boldsymbol{x},t)\Big) &= \int\limits_{\mathcal{V}_\mathrm{f}(t)} \boldsymbol{a}_\mathrm{f}(\boldsymbol{y},t)\cdot\boldsymbol{\nabla}_{\boldsymbol{x}}[w(\boldsymbol{x}-\boldsymbol{y})]\,\mathrm{d}V_y\\ &= -\int\limits_{\mathcal{V}_\mathrm{f}(t)} \boldsymbol{a}_\mathrm{f}(\boldsymbol{y},t)\cdot\boldsymbol{\nabla}_{\boldsymbol{y}}[w(\boldsymbol{x}-\boldsymbol{y})]\,\mathrm{d}V_y\end{aligned} \tag{3.2.31}$$

and apply the product rule and the divergence theorem to obtain

$$\begin{aligned}\boldsymbol{\nabla}\cdot\Big(\varepsilon_\mathrm{f}(\boldsymbol{x},t)\langle\boldsymbol{a}_\mathrm{f}\rangle^\mathrm{f}(\boldsymbol{x},t)\Big) &= \int\limits_{\mathcal{V}_\mathrm{f}(t)} (\boldsymbol{\nabla}\cdot\boldsymbol{a}_\mathrm{f})(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y\\ &\quad - \int\limits_{\partial\mathcal{V}_\mathrm{f}(t)} [\boldsymbol{a}_\mathrm{f}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})]\cdot\boldsymbol{n}_\mathrm{f}\,\mathrm{d}S_y\end{aligned} \tag{3.2.32}$$

With the same arguments as before, this leads to the averaging theorem in the form

$$\varepsilon_\mathrm{f}\langle\boldsymbol{\nabla}\cdot\boldsymbol{a}_\mathrm{f}\rangle^\mathrm{f} = \boldsymbol{\nabla}\cdot\Big(\varepsilon_\mathrm{f}\langle\boldsymbol{a}_\mathrm{f}\rangle^\mathrm{f}\Big) - \sum_{i=1}^{N_\mathrm{s}} \int\limits_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x}-\boldsymbol{y})\boldsymbol{a}_\mathrm{f}(\boldsymbol{y},t)\cdot\boldsymbol{n}_\mathrm{s}\,\mathrm{d}S_y \tag{3.2.33}$$

3.2.2 Mass conservation

Now that we have provided a way to reformulate the integral terms, we want to derive averaged versions of the Navier-Stokes equations. Although we will later restrict ourselves to an incompressible continuous phase, we do not incorporate this into the derivation of mass and momentum conservation in term of averaged quantities. This is done in order to provide a framework for possible future extensions. To improve readability, superscripts denoting the averaging volume will be omitted during the derivations, whenever the averaged quantity is clearly attached to only one of the phases. Let us first consider the point-wise mass conservation equation

$$\frac{\partial \rho_\mathrm{f}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}) = 0 \tag{3.2.34}$$

For both terms in this equation, we apply a convolution with the weighting function over the fluid volume, which leads to

$$\int\limits_{\mathcal{V}_\mathrm{f}(t)} \frac{\partial \rho_\mathrm{f}}{\partial t}(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) \,\mathrm{d}V_y + \int\limits_{\mathcal{V}_\mathrm{f}(t)} \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f})(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) \,\mathrm{d}V_y = 0 \tag{3.2.35}$$

or in short form

$$\varepsilon_\mathrm{f} \langle \frac{\partial \rho_\mathrm{f}}{\partial t} \rangle + \varepsilon_\mathrm{f} \langle \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}) \rangle = 0 \tag{3.2.36}$$

Setting $\psi_\mathrm{f} = \rho_\mathrm{f}$ in (3.2.29) we directly obtain

$$\varepsilon_\mathrm{f} \langle \frac{\partial \rho_\mathrm{f}}{\partial t} \rangle = \frac{\partial}{\partial t}(\varepsilon_\mathrm{f} \langle \rho_\mathrm{f} \rangle) + \sum_{i=1}^{N_\mathrm{s}} \int\limits_{S_{\mathrm{p},i}(t)} \rho_\mathrm{f}(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) (\boldsymbol{v}_{\mathrm{s},i} \cdot \boldsymbol{n}_\mathrm{s}) \,\mathrm{d}S_y \tag{3.2.37}$$

while for the second term in (3.2.36) we use (3.2.33) with $\boldsymbol{a}_\mathrm{f} = \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}$, yielding

$$\varepsilon_\mathrm{f} \langle \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}) \rangle = \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{f} \langle \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \rangle) - \sum_{i=1}^{N_\mathrm{s}} \int\limits_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x} - \boldsymbol{y}) \rho_\mathrm{f}(\boldsymbol{y}, t) (\boldsymbol{u}_\mathrm{f}(\boldsymbol{y}, t) \cdot \boldsymbol{n}_\mathrm{s}) \,\mathrm{d}S_y \tag{3.2.38}$$

As discussed before, the fluid velocity on the particle surface satisfies $\boldsymbol{u}_\mathrm{f} \cdot \boldsymbol{n}_\mathrm{s} = \boldsymbol{v}_{\mathrm{s},i} \cdot \boldsymbol{n}_\mathrm{s}$, such that the sums on the right side of (3.2.37) as well as (3.2.37) cancel out and we obtain

$$\boxed{\frac{\partial \left(\varepsilon_\mathrm{f} \langle \rho_\mathrm{f} \rangle^\mathrm{f} \right)}{\partial t} + \boldsymbol{\nabla} \cdot \left(\varepsilon_\mathrm{f} \langle \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \rangle^\mathrm{f} \right) = 0} \tag{3.2.39}$$

Our goal is to obtain a formulation where only averages of individual quantities are taken, which means that we need to reformulate the term $\varepsilon_\mathrm{f} \langle \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \rangle$ inside the divergence. If we consider an incompressible fluid, i.e. $\rho_\mathrm{f} = const$, this becomes trivial since $\langle \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \rangle = \rho_\mathrm{f} \langle \boldsymbol{u}_\mathrm{f} \rangle$, while for variable density the analogous formula $\langle \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \rangle = \langle \rho_\mathrm{f} \rangle \langle \boldsymbol{u}_\mathrm{f} \rangle$ is not correct and does in fact neglect small scale contributions. A common way to approach this problem in the context of turbulence modeling is the introduction of *Favre filtering* [50, 53], also called *mass averaging* [26]. The filtered quantity is then defined by

$$\langle \boldsymbol{a}_\mathrm{f} \rangle_\mathrm{F} = \frac{\langle \rho_\mathrm{f} \boldsymbol{a}_\mathrm{f} \rangle}{\langle \rho_\mathrm{f} \rangle} \tag{3.2.40}$$

and in analogy to the previous averaging, we write

$$\boldsymbol{a}_\mathrm{f} = \langle \boldsymbol{a}_\mathrm{f} \rangle_\mathrm{F} + \boldsymbol{a}_\mathrm{f}'' \tag{3.2.41}$$

with $\langle \boldsymbol{a}_{\mathrm{f}}'' \rangle_{\mathrm{F}} = 0$. Note that the Favre filtering reduces to the original averaging for a spatially non-varying density and that an analogous definition with volume fraction instead of material density leads to the notion of *phase averages* [50]. With this definition we can rewrite the averaged mass conservation into

$$\frac{\partial\left(\varepsilon_{\mathrm{f}}\langle\rho_{\mathrm{f}}\rangle^{\mathrm{f}}\right)}{\partial t} + \boldsymbol{\nabla}\cdot\left(\varepsilon_{\mathrm{f}}\langle\rho_{\mathrm{f}}\rangle^{\mathrm{f}}\langle\boldsymbol{u}_{\mathrm{f}}\rangle_{\mathrm{F}}^{\mathrm{f}}\right) = 0 \tag{3.2.42}$$

which, analogously to the single phase equation (1.4.4), can be reformulated into the Lagrangian form

$$\boxed{\frac{\mathrm{D}\left(\varepsilon_{\mathrm{f}}\langle\rho_{\mathrm{f}}\rangle^{\mathrm{f}}\right)}{\mathrm{D}t} + \varepsilon_{\mathrm{f}}\langle\rho_{\mathrm{f}}\rangle^{\mathrm{f}}\left(\boldsymbol{\nabla}\cdot\langle\boldsymbol{u}_{\mathrm{f}}\rangle_{\mathrm{F}}^{\mathrm{f}}\right) = 0} \tag{3.2.43}$$

Before proceeding with the momentum conservation, we consider the mass conservation in the solid phase. Let $\boldsymbol{u}_{\mathrm{s}} : \mathcal{V}_{\mathrm{s}} \times \mathbb{R}_+ \to \mathbb{R}^d$ denote the pointwise continuum velocity inside the solid phase. Inside each particle we can formulate the same point-wise mass conservation

$$\frac{\partial\rho_{\mathrm{s}}}{\partial t} + \boldsymbol{\nabla}\cdot(\rho_{\mathrm{s}}\boldsymbol{u}_{\mathrm{s}}) = 0 \tag{3.2.44}$$

as in the fluid phase. Another convolution with the weighting function, this time over the solid volume, followed by the application of product rule and Reynolds transport theorem results in

$$\begin{aligned}
&\int\limits_{\mathcal{V}_{\mathrm{p},i}(t)} \frac{\partial\rho_{\mathrm{s}}}{\partial t}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y \\
&= \frac{\partial}{\partial t}\int\limits_{\mathcal{V}_{\mathrm{p},i}(t)} \rho_{\mathrm{s}}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y - \int\limits_{S_{\mathrm{s},i}(t)} \rho_{\mathrm{s}}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{v}^{\mathrm{b}}\cdot\boldsymbol{n})\,\mathrm{d}S_y
\end{aligned} \tag{3.2.45}$$

for the first term in (3.2.44) and

$$\begin{aligned}
&\int\limits_{\mathcal{V}_{\mathrm{p},i}(t)} \nabla_{\boldsymbol{y}}\cdot(\rho_{\mathrm{s}}\boldsymbol{u}_{\mathrm{s}})(\boldsymbol{y},t)(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y \\
&= \nabla_{\boldsymbol{x}}\cdot\int\limits_{\mathcal{V}_{\mathrm{p},i}(t)} \rho_{\mathrm{s}}\boldsymbol{u}_{\mathrm{s}}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y - \int\limits_{S_{\mathrm{s},i}(t)} \rho_{\mathrm{s}}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{u}_{\mathrm{s}}\cdot\boldsymbol{n})\,\mathrm{d}S_y
\end{aligned} \tag{3.2.46}$$

for the second one. The surface velocity $\boldsymbol{v}^{\mathrm{b}}$ is identical to the solid continuum velocity $\boldsymbol{u}_{\mathrm{s}}$ such that the surface integrals cancel out upon summation and in analogy to (3.2.39) we obtain

$$\boxed{\frac{\partial(\varepsilon_{\mathrm{s}}\langle\rho_{\mathrm{s}}\rangle^{\mathrm{s}})}{\partial t} + \boldsymbol{\nabla}\cdot(\varepsilon_{\mathrm{s}}\langle\rho_{\mathrm{s}}\boldsymbol{u}_{\mathrm{s}}\rangle^{\mathrm{s}}) = 0} \tag{3.2.47}$$

Note that if we combine the averaged mass conservation equations of both phases, we get

$$\frac{\partial\left(\varepsilon_{\mathrm{f}}\langle\rho_{\mathrm{f}}\rangle^{\mathrm{f}} + \varepsilon_{\mathrm{s}}\langle\rho_{\mathrm{s}}\rangle^{\mathrm{s}}\right)}{\partial t} + \boldsymbol{\nabla}\cdot\left(\varepsilon_{\mathrm{f}}\langle\rho_{\mathrm{f}}\boldsymbol{u}_{\mathrm{f}}\rangle^{\mathrm{f}} + \varepsilon_{\mathrm{s}}\langle\rho_{\mathrm{s}}\boldsymbol{u}_{\mathrm{s}}\rangle^{\mathrm{s}}\right) = 0 \tag{3.2.48}$$

To reformulate this in a notation commonly used in multiphase flows, we introduce the *total volume average*

$$\langle\psi\rangle^{\mathrm{t}}(\boldsymbol{x},t) = \varepsilon_{\mathrm{f}}(\boldsymbol{x},t)\langle\psi\rangle^{\mathrm{f}}(\boldsymbol{x},t) + \varepsilon_{\mathrm{s}}(\boldsymbol{x},t)\langle\psi\rangle^{\mathrm{s}}(\boldsymbol{x},t) \tag{3.2.49}$$

defined for quantities that exist in both phases. With this, the above equation reads

$$\frac{\partial \langle \rho \rangle^{\mathrm{t}}}{\partial t} + \boldsymbol{\nabla} \cdot \langle \rho \boldsymbol{u} \rangle^{\mathrm{t}} = 0 \tag{3.2.50}$$

If both phases have constant density, it can be eliminated from both mass conservation equations and we can use (3.2.24) to obtain

$$\boldsymbol{\nabla} \cdot \left(\varepsilon_{\mathrm{s}} \langle \boldsymbol{u}_{\mathrm{s}} \rangle^{\mathrm{s}} + \varepsilon_{\mathrm{f}} \langle \boldsymbol{u}_{\mathrm{f}} \rangle^{\mathrm{f}} \right) = 0 \tag{3.2.51}$$

which is also sometimes used in the CFD-DEM context [150]. Considering the nature of our numerical framework, it might be tempting at this point, to rewrite the solid phase mass conservation into a Lagrangian formulation and combine it with that of the fluid phase. The possible pitfall in this is the fact that, the material derivative notation does not sufficiently represent the convection velocity. So one might accidentally miss the fact that in general

$$\frac{\mathrm{D}\left(\varepsilon_{\mathrm{f}} \langle \rho_{\mathrm{f}} \rangle^{\mathrm{f}} \right)}{\mathrm{D}t} + \frac{\mathrm{D}(\varepsilon_{\mathrm{s}} \langle \rho_{\mathrm{s}} \rangle^{\mathrm{s}})}{\mathrm{D}t} \neq \frac{\mathrm{D}\left(\varepsilon_{\mathrm{f}} \langle \rho_{\mathrm{f}} \rangle^{\mathrm{f}} + \varepsilon_{\mathrm{s}} \langle \rho_{\mathrm{s}} \rangle^{\mathrm{s}} \right)}{\mathrm{D}t} \tag{3.2.52}$$

where the right-hand side vanishes for constant densities. Combining the Lagrangian mass conservation equations and using this erroneous equality yields a misleading equation for the divergence of averaged fluid velocity.

3.2.3 Momentum conservation

Next, we want to apply the same kind of averaging to the point-wise momentum conservation equation of the fluid phase

$$\frac{\partial (\rho_{\mathrm{f}} \boldsymbol{u}_{\mathrm{f}})}{\partial t} + \boldsymbol{\nabla} \cdot (\rho_{\mathrm{f}} \boldsymbol{u}_{\mathrm{f}} \otimes \boldsymbol{u}_{\mathrm{f}}) = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_{\mathrm{f}} + \rho_{\mathrm{f}} \boldsymbol{g} \tag{3.2.53}$$

After integrating over the time derivative we can again apply (3.2.29), this time component wise with $\psi_{\mathrm{f}} = \rho_{\mathrm{f}} u_{\mathrm{f}}^{(j)}$, to obtain

$$\begin{aligned} &\int_{\mathcal{V}_{\mathrm{f}}(t)} \frac{\partial (\rho_{\mathrm{f}} \boldsymbol{u}_{\mathrm{f}})}{\partial t}(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) \, \mathrm{d}V_y \\ &= \frac{\partial (\varepsilon_{\mathrm{f}} \langle \rho_{\mathrm{f}} \boldsymbol{u}_{\mathrm{f}} \rangle)}{\partial t} + \sum_{i=1}^{N_{\mathrm{s}}} \int_{S_{\mathrm{p},i}(t)} \rho_{\mathrm{f}} \boldsymbol{u}_{\mathrm{f}}(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) (\boldsymbol{v}_{\mathrm{s},i} \cdot \boldsymbol{n}_{\mathrm{s}}) \, \mathrm{d}S_y \end{aligned} \tag{3.2.54}$$

Next, we consider the integral over the j-th component of the convective term and use (3.2.33) with $\boldsymbol{a}_{\mathrm{f}} = \rho_{\mathrm{f}} u_{\mathrm{f}}^{(j)} \boldsymbol{u}_{\mathrm{f}}$ to obtain

$$\begin{aligned} &\int_{\mathcal{V}_{\mathrm{f}}(t)} \boldsymbol{\nabla} \cdot \left(\rho_{\mathrm{f}} u_{\mathrm{f}}^{(j)} \boldsymbol{u}_{\mathrm{f}} \right)(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) \, \mathrm{d}V_y \\ &= \boldsymbol{\nabla} \cdot \left(\varepsilon_{\mathrm{f}} \langle \rho_{\mathrm{f}} u_{\mathrm{f}}^{(j)} \boldsymbol{u}_{\mathrm{f}} \rangle \right)(\boldsymbol{x}, t) - \sum_{i=1}^{N_{\mathrm{s}}} \int_{S_{\mathrm{p},i}(t)} \rho_{\mathrm{f}} u_{\mathrm{f}}^{(j)}(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) (\boldsymbol{v}_{\mathrm{s},i} \cdot \boldsymbol{n}_{\mathrm{s}}) \, \mathrm{d}S_y \end{aligned} \tag{3.2.55}$$

To rewrite the divergence in terms of individual averages, we first use the splitting (3.2.41), i.e.

$$\langle \rho_{\mathrm{f}} u_{\mathrm{f}}^{(j)} \boldsymbol{u}_{\mathrm{f}} \rangle = \langle \rho_{\mathrm{f}} \langle u_{\mathrm{f}} \rangle_{\mathrm{F}}^{(j)} \langle \boldsymbol{u}_{\mathrm{f}} \rangle_{\mathrm{F}} \rangle + \langle \rho_{\mathrm{f}} \langle u_{\mathrm{f}} \rangle_{\mathrm{F}}^{(j)} \boldsymbol{u}_{\mathrm{f}}'' \rangle + \langle \rho_{\mathrm{f}} u_{\mathrm{f}}''^{,(j)} \langle \boldsymbol{u}_{\mathrm{f}} \rangle_{\mathrm{F}} \rangle + \langle \rho_{\mathrm{f}} u_{\mathrm{f}}''^{,(j)} \boldsymbol{u}_{\mathrm{f}}'' \rangle \tag{3.2.56}$$

From the separation of scales assumption and the averaging definitions, we infer that average values vary little on the characteristic scale of the weighting function and can thus be taken outside the integration such that

$$\langle \rho_\mathrm{f} u_\mathrm{f}^{(j)} \boldsymbol{u}_\mathrm{f} \rangle = \langle u_\mathrm{f} \rangle_\mathrm{F}^{(j)} \langle \boldsymbol{u}_\mathrm{f} \rangle_\mathrm{F} \langle \rho_\mathrm{f} \rangle + \langle u_\mathrm{f} \rangle_\mathrm{F}^{(j)} \langle \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}'' \rangle + \langle \rho_\mathrm{f} u_\mathrm{f}''^{,(j)} \rangle \langle \boldsymbol{u}_\mathrm{f} \rangle_\mathrm{F} + \langle \rho_\mathrm{f} u_\mathrm{f}''^{,(j)} \boldsymbol{u}_\mathrm{f}'' \rangle \tag{3.2.57}$$

If we use $\langle \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}'' \rangle = \langle \rho_\mathrm{f} \rangle \langle \boldsymbol{u}_\mathrm{f}'' \rangle_\mathrm{F}$ and $\langle \boldsymbol{u}_\mathrm{f}'' \rangle_\mathrm{F} = 0$, this simplifies to

$$\langle \rho_\mathrm{f} u_\mathrm{f}^{(j)} \boldsymbol{u}_\mathrm{f} \rangle = \langle u_\mathrm{f} \rangle_\mathrm{F}^{(j)} \langle \boldsymbol{u}_\mathrm{f} \rangle_\mathrm{F} \langle \rho_\mathrm{f} \rangle + \langle \rho_\mathrm{f} u_\mathrm{f}''^{,(j)} \boldsymbol{u}_\mathrm{f}'' \rangle \tag{3.2.58}$$

and by collecting the results for the row-wise tensor divergence we can write the full convective term integral as

$$\begin{aligned} &\int_{\mathcal{V}_\mathrm{f}(t)} \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \otimes \boldsymbol{u}_\mathrm{f})(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) \, \mathrm{d}V_y = \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{f} \langle \rho_\mathrm{f} \rangle \langle \boldsymbol{u}_\mathrm{f} \rangle_\mathrm{F} \otimes \langle \boldsymbol{u}_\mathrm{f} \rangle_\mathrm{F}) \\ &+ \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{f} \langle \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}'' \otimes \boldsymbol{u}_\mathrm{f}'' \rangle) - \sum_{i=1}^{N_\mathrm{s}} \int_{S_{\mathrm{p},i}(t)} \rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) (\boldsymbol{v}_{\mathrm{s},i} \cdot \boldsymbol{n}_\mathrm{s}) \, \mathrm{d}S_y \end{aligned} \tag{3.2.59}$$

Adding this to the result obtained for the momentum density time derivative (3.2.54), the averaged left-hand side of the point wise momentum equation takes the form

$$\begin{aligned} &\int_{\mathcal{V}_\mathrm{f}(t)} \left[\frac{\partial (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f})}{\partial t} + \boldsymbol{\nabla} \cdot (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f} \otimes \boldsymbol{u}_\mathrm{f}) \right] (\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) \, \mathrm{d}V_y \\ &= \frac{\partial (\varepsilon_\mathrm{f} \langle \rho_\mathrm{f} \rangle \langle \boldsymbol{u}_\mathrm{f} \rangle_\mathrm{F})}{\partial t} + \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{f} \langle \rho_\mathrm{f} \rangle \langle \boldsymbol{u}_\mathrm{f} \rangle_\mathrm{F} \otimes \langle \boldsymbol{u}_\mathrm{f} \rangle_\mathrm{F}) + \boldsymbol{\nabla} \cdot \mathbf{R}_\mathbf{u} \end{aligned} \tag{3.2.60}$$

In this reformulation

$$\boxed{\mathbf{R}_\mathbf{u} = \int_{\mathcal{V}_\mathrm{f}(t)} (\rho_\mathrm{f} \boldsymbol{u}_\mathrm{f}'' \otimes \boldsymbol{u}_\mathrm{f}'')(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) \, \mathrm{d}V_y} \tag{3.2.61}$$

is the *residual stress tensor* which is analogous to the Reynolds stresses obtained by temporal averaging of the Navier-Stokes equations in single-phase turbulent flow. It is effected not only by the single-phase turbulence but also by the spatial variations in velocity due to the presence of particles.

Now that we have transferred the left-hand side into an expression in averaged quantities and a residual term, we want to turn our attention towards the remaining terms on the right-hand side. For the stress tensor, componentwise application of (3.2.33) leads to

$$\int_{\mathcal{V}_\mathrm{f}(t)} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{y}, t) w(\boldsymbol{x} - \boldsymbol{y}) \, \mathrm{d}V_y = \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{f} \langle \boldsymbol{\sigma}_\mathrm{f} \rangle) - \sum_{i=1}^{N_\mathrm{s}} \int_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x} - \boldsymbol{y}) (\boldsymbol{n}_\mathrm{s} \cdot \boldsymbol{\sigma}_\mathrm{f}) \, \mathrm{d}S_y \tag{3.2.62}$$

To analyze the integrals over particle surfaces, we split the stress tensor into its average and fluctuating components, i.e.

$$\begin{aligned} &\int_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x} - \boldsymbol{y}) (\boldsymbol{n}_\mathrm{s} \cdot \boldsymbol{\sigma}_\mathrm{f}) \, \mathrm{d}S_y \\ &= \int_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x} - \boldsymbol{y}) (\boldsymbol{n}_\mathrm{s} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle) \, \mathrm{d}S_y + \int_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x} - \boldsymbol{y}) (\boldsymbol{n}_\mathrm{s} \cdot \boldsymbol{\sigma}_\mathrm{f}') \, \mathrm{d}S_y \end{aligned} \tag{3.2.63}$$

First, we want to rewrite the integral over the averaged stress tensor. We note that $\langle \boldsymbol{\sigma}_\mathrm{f} \rangle$ is defined on the whole domain, including the solid phase, such that we can apply the divergence theorem on individual particle volumes. Thus

$$\int\limits_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{n}_\mathrm{s} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle)\,\mathrm{d}S_y = \int\limits_{\mathcal{V}_{\mathrm{p},i}(t)} \boldsymbol{\nabla}_{\boldsymbol{y}} \cdot (\langle \boldsymbol{\sigma}_\mathrm{f} \rangle(\boldsymbol{y},t) w(\boldsymbol{x}-\boldsymbol{y}))\,\mathrm{d}V_y \tag{3.2.64}$$

Applying the product rule and changing the argument of differentiation for the weighting function gradient, we obtain

$$\begin{aligned} &\int\limits_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{n}_\mathrm{s} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle)\,\mathrm{d}S_y \\ &= \int\limits_{\mathcal{V}_{\mathrm{p},i}(t)} (\boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle)(\boldsymbol{y},t) w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y - \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \int\limits_{\mathcal{V}_{\mathrm{p},i}(t)} \langle \boldsymbol{\sigma}_\mathrm{f} \rangle(\boldsymbol{y},t) w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y \end{aligned} \tag{3.2.65}$$

Summing up the contributions of all particles and recalling the definition of solid phase averages we can write

$$\sum_{i=1}^{N_\mathrm{s}} \int\limits_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{n}_\mathrm{s} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle)\,\mathrm{d}S_y = \varepsilon_\mathrm{s} \langle \boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle^\mathrm{f} \rangle^\mathrm{s} - \boldsymbol{\nabla} \cdot \left(\varepsilon_\mathrm{s} \langle \langle \boldsymbol{\sigma}_\mathrm{f} \rangle^\mathrm{f} \rangle^\mathrm{s} \right) \tag{3.2.66}$$

With the assumption (3.2.20), the solid phase average in the second term can be dropped and by plugging (3.2.63) and (3.2.66) into (3.2.62) we obtain

$$\begin{aligned} &\int\limits_{\mathcal{V}_\mathrm{f}(t)} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{y},t) w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y \\ &= \boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle - \varepsilon_\mathrm{s} \langle \boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle \rangle^\mathrm{s} - \sum_{i=1}^{N_\mathrm{s}} \int\limits_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{n}_\mathrm{s} \cdot \boldsymbol{\sigma}'_\mathrm{f})\,\mathrm{d}S_y \end{aligned} \tag{3.2.67}$$

where we used

$$\boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{f} \langle \boldsymbol{\sigma}_\mathrm{f} \rangle) + \boldsymbol{\nabla} \cdot (\varepsilon_\mathrm{s} \langle \boldsymbol{\sigma}_\mathrm{f} \rangle) = \boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle \tag{3.2.68}$$

Next, we want to show that the second and third term in (3.2.67) represent the momentum exchange between the phases and can be written in terms of weighted averages over forces onto individual particles. For this, let us consider the total force exerted by the fluid phase onto a particle, i.e.

$$\boldsymbol{F}_{\mathrm{fs},i} = \int\limits_{S_{\mathrm{p},i}} \boldsymbol{\sigma}_\mathrm{f} \boldsymbol{n}_\mathrm{s}\,\mathrm{d}S = \int\limits_{\mathcal{V}_{\mathrm{p},i}} \boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle\,\mathrm{d}V + \int\limits_{S_{\mathrm{p},i}} \boldsymbol{n}_\mathrm{s} \cdot \boldsymbol{\sigma}'_\mathrm{f}\,\mathrm{d}S \tag{3.2.69}$$

We directly observe, that the field $\boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle$, defined on both, the fluid and solid phase, is a body force volume density acting on the solid phase. We further recognize this term from (3.2.67), where it is averaged over the whole solid phase and contributes to the fluid momentum with a negative sign. If we assume that both, $\boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle$ and the weighting function, are constant within a single particle volume, we can simplify

$$\varepsilon_\mathrm{s} \langle \boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle \rangle^\mathrm{s} \approx \sum_{i=1}^{N_\mathrm{s}} \boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle(\boldsymbol{x}_{\mathrm{s},i}) V_{\mathrm{s},i} w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i}) \tag{3.2.70}$$

$$\int\limits_{\mathcal{V}_{\mathrm{p},i}} \boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle\,\mathrm{d}V \approx \boldsymbol{\nabla} \cdot \langle \boldsymbol{\sigma}_\mathrm{f} \rangle(\boldsymbol{x}_{\mathrm{s},i}) V_{\mathrm{s},i} \tag{3.2.71}$$

Note that, in implementations of the unresolved CFD-DEM approach, this assumption has to be made anyway, because the discretization of the continuous phase typically doesn't allow a better approximation inside a single particle volume. Let us further assume that

$$\int\limits_{S_{\mathrm{p},i}(t)} w(\boldsymbol{x}-\boldsymbol{y})(\boldsymbol{n}_\mathrm{s}\cdot\boldsymbol{\sigma}'_\mathrm{f})\,\mathrm{d}S_y \approx w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i}) \int\limits_{S_{\mathrm{p},i}(t)} \boldsymbol{n}_\mathrm{s}\cdot\boldsymbol{\sigma}'_\mathrm{f}\,\mathrm{d}S_y \tag{3.2.72}$$

is a valid approximation. Combined with the simplifications (3.2.70) and (3.2.71), this allows us to write

$$\int\limits_{\mathcal{V}_\mathrm{f}(t)} \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{y},t)w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y = \boldsymbol{\nabla}\cdot\langle\boldsymbol{\sigma}_\mathrm{f}\rangle - \boldsymbol{f}_\mathrm{fs} \tag{3.2.73}$$

where

$$\boldsymbol{f}_\mathrm{fs}(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{s}} w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i})\boldsymbol{F}_{\mathrm{fs},i} \tag{3.2.74}$$

Here, the forces onto individual particles are approximated, as discussed above, by

$$\boldsymbol{F}_{\mathrm{fs},i} = V_{\mathrm{s},i}\Big(\boldsymbol{\nabla}\cdot\langle\boldsymbol{\sigma}_\mathrm{f}\rangle^\mathrm{f}\Big)(\boldsymbol{x}_{\mathrm{s},i}) + \int\limits_{S_{\mathrm{p},i}} \boldsymbol{n}_\mathrm{s}\cdot\boldsymbol{\sigma}'_\mathrm{f}\,\mathrm{d}S \tag{3.2.75}$$

Anderson and Jackson [4] have shown that an identity of the form

$$\int\limits_{S_{\mathrm{p},i}} \psi(\boldsymbol{y})n_k(\boldsymbol{y})w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}S \approx w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i}) \int\limits_{S_{\mathrm{p},i}} \psi(\boldsymbol{y})n_k(\boldsymbol{y})\,\mathrm{d}S \tag{3.2.76}$$

results in a good approximation when the following requirements are met

- The mean value of ψ_f on the particle surface $S_{\mathrm{p},i}$ is small when compared to the variations of this quantity about its mean
- The scalar field ψ_f varies smoothly over the surface $S_{\mathrm{p},i}$
- The particle surface $S_{\mathrm{p},i}$ is approximately spherical

As is common practice not only within the derivation of volume averaged equations but also in applications of unresolved CFD-DEM method, we consider perfectly spherical particles within this thesis, such that the third point is satisfied. According to Anderson and Jackson [4] the remaining requirements are also fulfilled for the components of the fluctuating stress tensor, such that (3.2.72) can be considered valid.

We also want to note that, since the coupling force $\boldsymbol{F}_{\mathrm{fs},i}$ is not defined throughout the solid volume but once for each particle, it is customary [72, 175] to write (3.2.74) with a different notion of average as

$$\boldsymbol{f}_\mathrm{fs}(\boldsymbol{x}) = n_\mathrm{p}(\boldsymbol{x})\langle\boldsymbol{F}_\mathrm{fs}\rangle^\mathrm{p}(\boldsymbol{x}) \tag{3.2.77}$$

In this identity n_p denotes the *particle number density*, i.e. the number of particles per unit volume, given by

$$n_\mathrm{p}(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{s}} w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i}) \tag{3.2.78}$$

and $\langle\cdot\rangle^\mathrm{p}$ denotes the *particle phase average* defined as

$$n_\mathrm{p}(\boldsymbol{x})\langle\psi\rangle^\mathrm{p}(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{s}} w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i})\psi_i \tag{3.2.79}$$

for a quantity ψ attached to the solid particles. To simplify an understanding of these definitions, we use the relation to moving averages via (3.2.13). It is then easy to see that n_p represents the number of particles within the averaging volume, while $\langle\psi\rangle^\mathrm{p}$ reduces to the conventional unweighted average over the values ψ_i of all particles within the averaging volume. Since we can write

$$\varepsilon_\mathrm{s}(\boldsymbol{x}) \approx \sum_{i=1}^{N_\mathrm{s}} w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i})V_{\mathrm{s},j} \tag{3.2.80}$$

it is evident that $\varepsilon_\mathrm{s} \approx n_\mathrm{p}V_\mathrm{s}$ for particles of identical volume $V_\mathrm{s} = V_{\mathrm{s},i},\ \forall i \in \{1,\dots,N_\mathrm{s}\}$. Additionally, if we have a quantity ψ which is constant within each particle, i.e.

$$\psi_\mathrm{s}(\boldsymbol{x}) = \sum_{j=1}^{N_\mathrm{s}} \psi_{\mathrm{s},j}\chi_{\mathcal{V}_{\mathrm{s},j}}(\boldsymbol{x}) \tag{3.2.81}$$

we obtain

$$\varepsilon_\mathrm{s}(\boldsymbol{x})\langle\psi_\mathrm{s}\rangle^\mathrm{s} \approx \sum_{i=1}^{N_\mathrm{s}} w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i})_{\mathrm{s},j}V_{\mathrm{s},j} \tag{3.2.82}$$

and observe that for particles of identical volume the two definitions of volume average and particle phase average coincide.

Getting back to reformulating the volume averaged right-hand side of (3.2.53), we notice that, after having dealt with the stress tensor integral, the only remaining part is due to body forces. Using mass averaging we obtain the additional term

$$\varepsilon_\mathrm{f}\langle\rho_\mathrm{f}\boldsymbol{k}\rangle = \varepsilon_\mathrm{f}\langle\rho_\mathrm{f}\rangle\langle\boldsymbol{k}\rangle_\mathrm{F} \tag{3.2.83}$$

If we further split the stress tensor, we obtain

$$\boldsymbol{\nabla}\cdot\langle\boldsymbol{\sigma}_\mathrm{f}\rangle = \boldsymbol{\nabla}\cdot\langle -p\mathbb{1}+\boldsymbol{\tau}_\mathrm{f}\rangle = -\boldsymbol{\nabla}\langle p\rangle + \boldsymbol{\nabla}\cdot\langle\boldsymbol{\tau}_\mathrm{f}\rangle \tag{3.2.84}$$

and the final form of the averaged momentum equations for the fluid phase reads

$$\begin{aligned}&\frac{\partial\left(\varepsilon_\mathrm{f}\langle\rho_\mathrm{f}\rangle^\mathrm{f}\langle\boldsymbol{u}_\mathrm{f}\rangle_\mathrm{F}^\mathrm{f}\right)}{\partial t} + \boldsymbol{\nabla}\cdot\left(\varepsilon_\mathrm{f}\langle\rho_\mathrm{f}\rangle^\mathrm{f}\langle\boldsymbol{u}_\mathrm{f}\rangle_\mathrm{F}^\mathrm{f}\otimes\langle\boldsymbol{u}_\mathrm{f}\rangle_\mathrm{F}^\mathrm{f}\right)\\ &= -\boldsymbol{\nabla}\langle p\rangle^\mathrm{f} + \boldsymbol{\nabla}\cdot\langle\boldsymbol{\tau}_\mathrm{f}\rangle^\mathrm{f} - \boldsymbol{\nabla}\cdot\mathbf{R}_\mathbf{u} - \boldsymbol{f}_\mathrm{fs} + \varepsilon_\mathrm{f}\langle\rho_\mathrm{f}\rangle^\mathrm{f}\langle\boldsymbol{k}\rangle_\mathrm{F}^\mathrm{f}\end{aligned} \tag{3.2.85}$$

Note that, by using the identity (1.5.10) and mass conservation (3.2.39) we can also write

$$\boxed{\varepsilon_\mathrm{f}\langle\rho_\mathrm{f}\rangle^\mathrm{f}\frac{D\langle\boldsymbol{u}_\mathrm{f}\rangle_\mathrm{F}^\mathrm{f}}{Dt} = -\boldsymbol{\nabla}\langle p\rangle^\mathrm{f} + \boldsymbol{\nabla}\cdot\langle\boldsymbol{\tau}_\mathrm{f}\rangle^\mathrm{f} - \boldsymbol{\nabla}\cdot\mathbf{R}_\mathbf{u} - \boldsymbol{f}_\mathrm{fs} + \varepsilon_\mathrm{f}\langle\rho_\mathrm{f}\rangle^\mathrm{f}\langle\boldsymbol{k}\rangle_\mathrm{F}^\mathrm{f}} \tag{3.2.86}$$

To close this set of equations we have to express the unclosed terms $\langle\boldsymbol{\tau}_\mathrm{f}\rangle^\mathrm{f}$, $\mathbf{R}_\mathbf{u}$ in terms of the volume fraction ε_f, averaged density $\langle\rho_\mathrm{f}\rangle$, Favre averaged fluid velocity field $\langle\boldsymbol{u}_\mathrm{f}\rangle^\mathrm{f}$ and average pressure $\langle p\rangle^\mathrm{f}$. This will be discussed in the next section.

At this point, we want to mention that one could proceed to derive an averaged version of the energy equation to treat compressible fluid phases (see e.g. [26] for a derivation based on averages without weighting). However, since we do restrict ourselves to the case of constant density in the continuous phase, we will not pursue this here.

Before diving into typical closure approaches, let us first try to reformulate the average stress tensor $\langle\boldsymbol{\tau}_\mathrm{f}\rangle$ for the Newtonian case in terms of averaged quantities and residual components below the averaging scale. For the point variables we consider the viscous stress tensor in the form of (1.5.16) and with constant coefficients of viscosity, i.e.

$$\langle\boldsymbol{\tau}_\mathrm{f}(\boldsymbol{u}_\mathrm{f})\rangle = \lambda\langle\boldsymbol{\nabla}\cdot\boldsymbol{u}_\mathrm{f}\rangle\mathbb{1} + \mu\Big(\langle\boldsymbol{\nabla}\boldsymbol{u}_\mathrm{f}\rangle + (\langle\boldsymbol{\nabla}\boldsymbol{u}_\mathrm{f}\rangle)^T\Big) \tag{3.2.87}$$

So the two terms we have to reformulate are $\langle \boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f} \rangle$ and $\langle \boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f} \rangle$. Using the splitting of point values into average and residual part, one can write

$$\langle \boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f} \rangle = \langle \boldsymbol{\nabla} \cdot \langle \boldsymbol{u}_\mathrm{f} \rangle \rangle + \langle \boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f}' \rangle = \boldsymbol{\nabla} \cdot \langle \boldsymbol{u}_\mathrm{f} \rangle - (\boldsymbol{\nabla} \cdot \langle \boldsymbol{u}_\mathrm{f} \rangle)' + \langle \boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f}' \rangle \tag{3.2.88}$$

$$\langle \boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f} \rangle = \boldsymbol{\nabla} \langle \boldsymbol{u}_\mathrm{f} \rangle - (\boldsymbol{\nabla} \langle \boldsymbol{u}_\mathrm{f} \rangle)' + \langle \boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f}' \rangle \tag{3.2.89}$$

such that the averaged viscous stress tensor becomes

$$\langle \boldsymbol{\tau}_\mathrm{f}(\boldsymbol{u}_\mathrm{f}) \rangle = \boldsymbol{\tau}_\mathrm{f}(\langle \boldsymbol{u}_\mathrm{f} \rangle) + \mathbf{R}_{\boldsymbol{\tau}_\mathrm{f}} \tag{3.2.90}$$

with residual stress tensor defined as

$$\mathbf{R}_{\boldsymbol{\tau}_\mathrm{f}} = \lambda \Big[\langle \boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f}' \rangle - (\boldsymbol{\nabla} \cdot \langle \boldsymbol{u}_\mathrm{f} \rangle)' \Big] \mathbb{1} + \mu \Bigg(\Big[\langle \boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f}' \rangle - (\boldsymbol{\nabla} \langle \boldsymbol{u}_\mathrm{f} \rangle)' \Big] + \Big[\langle \boldsymbol{\nabla} \boldsymbol{u}_\mathrm{f}' \rangle - (\boldsymbol{\nabla} \langle \boldsymbol{u}_\mathrm{f} \rangle)' \Big]^T \Bigg) \tag{3.2.91}$$

This tensor is another quantity which formally needs a closure.

3.2.4 Closures

In an attempt to provide a simple expression for the stresses, Anderson and Jackson [4] proposed one of the first closures by subsuming all stresses in

$$\langle \boldsymbol{\Pi} \rangle = \langle \boldsymbol{\sigma}_\mathrm{f} \rangle - \langle \mathbf{R}_\mathbf{u} \rangle \tag{3.2.92}$$

and making the assumption, that the stress tensor $\langle \boldsymbol{\Pi} \rangle$ can be expressed analogously to that of the point equations. For a Newtonian fluid they proposed the usage of

$$\langle \boldsymbol{\Pi} \rangle = -\langle p \rangle \mathbb{1} + \lambda(\varepsilon_\mathrm{f})(\boldsymbol{\nabla} \cdot \langle \boldsymbol{u}_\mathrm{f} \rangle) \mathbb{1} + \mu(\varepsilon_\mathrm{f}) \Big[\boldsymbol{\nabla} \langle \boldsymbol{u}_\mathrm{f} \rangle + (\boldsymbol{\nabla} \langle \boldsymbol{u}_\mathrm{f} \rangle)^T - \frac{2}{3} (\boldsymbol{\nabla} \cdot \langle \boldsymbol{u}_\mathrm{f} \rangle) \mathbb{1} \Big] \tag{3.2.93}$$

where $\lambda(\varepsilon_\mathrm{f})$ and $\mu(\varepsilon_\mathrm{f})$ denote the effective bulk and dynamic viscosities and have to be chosen in such a way as to model both, the stresses from $\langle \boldsymbol{\sigma}_\mathrm{f} \rangle$ as well as $\langle \mathbf{R}_\mathbf{u} \rangle$. Since then, a lot of closures have been proposed which model both terms separately.

In [14] different closures for the residual viscous stresses $\mathbf{R}_{\boldsymbol{\tau}_\mathrm{f}}$ are discussed. The model that the authors apply is again based on the idea of modeling the Reynolds term in the same form as the viscous stresses and employing an effective viscosity. For this type of model the authors refer to [114, 171] and use the effective viscosity derived in [54] for fluidized beds.

Similarly, for the Reynolds type stress from the left-hand side $\mathbf{R}_\mathbf{u}$ there are a variety of closures. Most of them are based on analogies to turbulence modeling in single phase flow (see e.g. [14, 26, 28, 102, 150]).

Nevertheless, it has become common practice in CFD-DEM to neglect both, $\mathbf{R}_{\boldsymbol{\tau}_\mathrm{f}}$ and $\mathbf{R}_\mathbf{u}$. Many numerical schemes are built on the resulting simplified model [52, 77, 168, 175] and it has been shown that good agreement with experimental results can still be obtained. Since neglecting both $\mathbf{R}_\mathbf{u}$ and $\mathbf{R}_{\boldsymbol{\tau}_\mathrm{f}}$ has been shown to be an acceptable approximation in the above mentioned literature, we also employ this simplification for the rest of this thesis. Extensions in this direction are kept in mind for future work after the framework has been sufficiently established and well tested. We further recall from Chapter 1 that we consider the viscous stress in volume viscosity formulation with Stokes hypothesis applied, so that the averaged viscous stress tensor is assumed to be of the form

$$\langle \boldsymbol{\tau}_\mathrm{f}(\boldsymbol{u}_\mathrm{f}) \rangle \approx \boldsymbol{\tau}_\mathrm{f}(\langle \boldsymbol{u}_\mathrm{f} \rangle) = \mu \Big(\boldsymbol{\nabla} \langle \boldsymbol{u}_\mathrm{f} \rangle + (\boldsymbol{\nabla} \langle \boldsymbol{u}_\mathrm{f} \rangle)^T - \frac{2}{3} (\boldsymbol{\nabla} \cdot \langle \boldsymbol{u}_\mathrm{f} \rangle) \mathbb{1} \Big) \tag{3.2.94}$$

Throughout this thesis we also consider the fluid phase to have constant material density. This reduces the Favre averages to fluid intrinsic averages and we finally obtain the volume averaged incompressible Navier-Stokes equations in the Lagrangian form

$$\frac{\mathrm{D}\varepsilon_\mathrm{f}}{\mathrm{D}t} = -\varepsilon_\mathrm{f}(\boldsymbol{\nabla}\cdot\langle\boldsymbol{u}_\mathrm{f}\rangle) \tag{3.2.95}$$

$$\varepsilon_\mathrm{f}\rho_\mathrm{f}\frac{D\langle\boldsymbol{u}_\mathrm{f}\rangle}{Dt} = -\boldsymbol{\nabla}\langle p\rangle + \boldsymbol{\nabla}\cdot\boldsymbol{\tau}_\mathrm{f}(\langle\boldsymbol{u}_\mathrm{f}\rangle) - \boldsymbol{f}_\mathrm{fs} + \varepsilon_\mathrm{f}\rho_\mathrm{f}\langle\boldsymbol{k}\rangle \tag{3.2.96}$$

For our numerical scheme we consider (3.2.95), (3.2.96) together with (3.2.94) and (3.2.74) as the set of equations we want to solve. We observe that by introducing the effective fluid density $\bar{\rho}_\mathrm{f} = \rho_\mathrm{f}\varepsilon_\mathrm{f}$ and multiplying the mass conservation with ρ_f, this is the same set of equations as for a single phase, now with a varying density and an additional momentum source term $\boldsymbol{f}_\mathrm{fs}$. For all problems considered within this thesis we further only consider body forces due to constant gravity so that $\langle\boldsymbol{k}\rangle = \boldsymbol{g}$ with gravitational acceleration $\boldsymbol{g}$.

3.3 Formulations in unresolved CFD-DEM

Now that we have clarified all necessary notations and derived a set of volume averaged equations we want to put them into the context of existing literature on unresolved CFD-DEM. We will see that there are slight differences which are not necessarily due to the above closures but due to the way that the coupling is realized in numerical schemes.

The first occurrence of unresolved CFD-DEM scheme in literature is commonly traced back to the work by Tsuji et al. [156, 157] in the early 90s. In [156] the authors also used a finite difference framework, albeit a Eulerian one, and considered volume averaged equations based on the work by Anderson and Jackson [4] for inviscid fluid flow. Since then, a huge number of numerical investigations based on the unresolved coupling have been performed for a variety of different applications. A detailed analysis of all these contributions would certainly go beyond the scope of this thesis and has already been done in several articles within the past two decades [48, 75, 175, 177]. Here we only want to concentrate on the different models based on the above set of equations (3.2.95), (3.2.96). Although not explicitly stated by the authors, references and equations in the review by Zhou et al. [175] show a restriction to constant density fluids and omission of residual stresses $\mathbf{R}_\mathbf{u}$, $\mathbf{R}_{\boldsymbol{\tau}_\mathrm{f}}$ leading to the same equations as the above derivation. In their article, they identified three different models which had been previously applied in literature (note that the authors partially use the notation introduced in (3.2.77)). All the considered models use the same mass conservation equation

$$\frac{\partial(\rho_\mathrm{f}\varepsilon_\mathrm{f})}{\partial t} + \boldsymbol{\nabla}\cdot(\rho_\mathrm{f}\varepsilon_\mathrm{f}\boldsymbol{u}_\mathrm{f}) = 0 \tag{3.3.1}$$

which is identical to what we previously derived. Note that we have additionally dropped the average notation for better readability. When considering the right-hand side of momentum equations, the models start to differ:

$$\text{Model I \& III:}\quad \frac{\partial(\rho_\mathrm{f}\varepsilon_\mathrm{f}\boldsymbol{u}_\mathrm{f})}{\partial t} + \boldsymbol{\nabla}\cdot(\rho_\mathrm{f}\varepsilon_\mathrm{f}\boldsymbol{u}_\mathrm{f}\otimes\boldsymbol{u}_\mathrm{f}) = -\boldsymbol{\nabla}p + \boldsymbol{\nabla}\cdot\boldsymbol{\tau}_\mathrm{f} - \boldsymbol{f}_\mathrm{fs}^{\mathrm{I,III}} + \rho_\mathrm{f}\varepsilon_\mathrm{f}\boldsymbol{g} \tag{3.3.2}$$

$$\text{Model II:}\quad \frac{\partial(\rho_\mathrm{f}\varepsilon_\mathrm{f}\boldsymbol{u}_\mathrm{f})}{\partial t} + \boldsymbol{\nabla}\cdot(\rho_\mathrm{f}\varepsilon_\mathrm{f}\boldsymbol{u}_\mathrm{f}\otimes\boldsymbol{u}_\mathrm{f}) = -\varepsilon_\mathrm{f}\boldsymbol{\nabla}p + \varepsilon_\mathrm{f}\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_\mathrm{f} - \boldsymbol{f}_\mathrm{fs}^{\mathrm{II}} + \rho_\mathrm{f}\varepsilon_\mathrm{f}\boldsymbol{g} \tag{3.3.3}$$

We observe that the general form of model I and III coincides with the result of our derivations in the previous section. The difference between these two models lies in the specific form of the forces onto solid particles. To understand this, we recall that the force $\boldsymbol{F}_{\mathrm{fs},i}$ onto a particle is split into a contribution due to the averaged fluid stress tensor evaluated at the

particle position and an integral in terms of unresolved stress fluctuations about the mean (see (3.2.75)). As is the case for the Reynolds type stress tensors, this integral needs to be modeled and expressed in terms of the available quantities, namely averaged fluid variables and properties of the individual particles. Employing the same notation as in [175], we write

$$\boldsymbol{F}_{\mathrm{fs},i} = \boldsymbol{F}_{\mathrm{p},i} + \boldsymbol{F}_{\tau,i} + \boldsymbol{F}_{\mathrm{d},i} + \boldsymbol{F}_i'' \tag{3.3.4}$$

where the fluid stress is split into a pressure and viscous stress term

$$\boldsymbol{F}_{\mathrm{p},i} = -\left(\boldsymbol{\nabla}\langle p\rangle\right)|_{\boldsymbol{x}_{\mathrm{s},i}} V_{\mathrm{s},i} \tag{3.3.5}$$

$$\boldsymbol{F}_{\tau,i} = \left(\boldsymbol{\nabla}\cdot\langle\boldsymbol{\tau}_{\mathrm{f}}\rangle\right)|_{\boldsymbol{x}_{\mathrm{s},i}} V_{\mathrm{s},i} \tag{3.3.6}$$

and the integral over small-scale fluctuations has been split into a drag force $\boldsymbol{F}_{\mathrm{d},i}$ and non-drag terms subsumed in $\boldsymbol{F}_i''$. This formulation reflects the observation that in many flow configurations the interaction between fluid and particles is dominated by drag. Following (3.2.74) the momentum source to the fluid phase is in our notation and for model I given by

$$\boldsymbol{f}_{\mathrm{fs}}^{\mathrm{I}}(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{s}}} w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i})(\boldsymbol{F}_{\mathrm{p},i} + \boldsymbol{F}_{\tau,i} + \boldsymbol{F}_{\mathrm{d},i} + \boldsymbol{F}_i'') \tag{3.3.7}$$

We note that the definition in [175] is slightly different since it is already tailored towards their choice of a mesh-based numerical scheme. Their notation results from taking the computational cells as averaging volumes. While this can not be directly obtained from the above weighting function formalism, since it requires a spatially varying weighting function shape, it bears the same meaning. More details on the calculation of averages in our numerical framework will be given at a later point, namely in Section 4.2. Proceeding with model III, we observe that the pressure and viscous stress forces are replaced by a buoyancy term and the forces due to stress fluctuations are scaled with the inverse fluid volume fraction so that

$$\boldsymbol{F}_{\mathrm{fs},i}^{\mathrm{III}} = \frac{\boldsymbol{F}_{\mathrm{d},i} + \boldsymbol{F}_i''}{\varepsilon_{\mathrm{s}}(\boldsymbol{x}_{\mathrm{s},i})} - \rho_{\mathrm{f}} V_{\mathrm{s},i}\boldsymbol{g} \tag{3.3.8}$$

and

$$\boldsymbol{f}_{\mathrm{fs}}^{\mathrm{III}}(\boldsymbol{x}) = \frac{1}{\varepsilon_{\mathrm{f}}(\boldsymbol{x})}\sum_{i=1}^{N_{\mathrm{s}}} w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i})(\boldsymbol{F}_{\mathrm{d},i} + \boldsymbol{F}_i'') - \sum_{i=1}^{N_{\mathrm{s}}} w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i})\rho_{\mathrm{f}} V_{\mathrm{s},i}\boldsymbol{g} \tag{3.3.9}$$

The applicability of model III and possible advantages over model I have been the subject of many articles, most notably in Kafui et al. [75] and Feng et al. [48] together with the corresponding published communications. However, Zhou et al. [175] do show that this formulation is based on a the simplifying assumption which technically requires a steady uniform flow in the fluid phase. Although they show that in gas-solid fluidized bed and pneumatic conveying, which are among the most prominent applications of CFD-DEM, this model can still provide very similar results, it fails to predict the correct behavior in more complex flows such as hydroclones. Due to this restriction and since there is also no clear advantage in choosing model III over the other two methods, we refrain from using this simplified model in our numerical scheme.

Instead, we only consider model I and II, which can be shown to be mathematically equivalent with the assumptions from the derivation of volume averaged equations. To realize this, we recall that, to derive model I, we made use of the assumption that $\boldsymbol{\nabla}\cdot\langle\boldsymbol{\sigma}_{\mathrm{f}}\rangle$ is constant within a single particle volume. Extending this to the assumption that the term has negligible variations within the characteristic length scale of the weighting function (as has been assumed for every averaged quantity) we can write

$$\varepsilon_{\mathrm{s}}\langle\boldsymbol{\nabla}\cdot\langle\boldsymbol{\sigma}_{\mathrm{f}}\rangle^{\mathrm{f}}\rangle^{\mathrm{s}} \approx \varepsilon_{\mathrm{s}}\Big(\boldsymbol{\nabla}\cdot\langle\boldsymbol{\sigma}_{\mathrm{f}}\rangle^{\mathrm{f}}\Big) \tag{3.3.10}$$

Further using the relation (3.2.70) we can write

$$\boldsymbol{f}^{\mathrm{I}}_{\mathrm{fs},i} = \varepsilon_{\mathrm{s}} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_{\mathrm{f}} + \boldsymbol{f}^{\mathrm{II}}_{\mathrm{fs},i} \tag{3.3.11}$$

$$\boldsymbol{f}^{\mathrm{II}}_{\mathrm{fs},i} = \sum_{i=1}^{N_{\mathrm{s}}} w(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},i})(\boldsymbol{F}_{\mathrm{d},i} + \boldsymbol{F}''_i) \tag{3.3.12}$$

in (3.3.2) and collect the stress tensor terms using $\varepsilon_{\mathrm{f}} = 1 - \varepsilon_{\mathrm{s}}$ to obtain (3.3.3). The equivalence of these two models has also been observed in the numerical results by Zhou et al. [175]. However, since our numerical scheme does not only deviate from the one in [175] due to its mesh free nature but also due to the chosen approach to solve the volume averaged equations, we will still consider both, model I and II, in the upcoming chapters and examine whether they lead to any significant differences in our framework.

Before we complete this discussion we want to note that, while the derivations in Section 3.2 were based on perfectly spherical particles, it has been shown that the above models of CFD-DEM with the same set of volume averaged Navier-Stokes equations can also be successfully applied to flows with non-spherical particles ([62, 174]). However, it is not always clear whether these extensions are based on a similarly solid mathematical foundation or if they are of heuristic nature and an assessment of this matter is not within the scope of the present work.

3.4 Coupling forces

In this section we want to shed some light on both, the expressions for the drag force $\boldsymbol{F}_{\mathrm{d}}$ and terms which are subsumed in the force $\boldsymbol{F}''$. Starting with the latter, a common form of $\boldsymbol{F}''_i$ [118] is

$$\boldsymbol{F}''_i = \boldsymbol{F}_{\mathrm{vm},i} + \boldsymbol{F}_{\mathrm{l},i} + \boldsymbol{F}_{\mathrm{h},i} \tag{3.4.1}$$

with $\boldsymbol{F}_{\mathrm{vm}}$, $\boldsymbol{F}_{\mathrm{l}}$, $\boldsymbol{F}_{\mathrm{h}}$ denoting the virtual mass force, lift force and history force acting on the particle. More specifically, it is customary to consider $\boldsymbol{F}_{\mathrm{l}} = \boldsymbol{F}_{\mathrm{Saff}} + \boldsymbol{F}_{\mathrm{Mag}}$ with Saffman and Magnus lift and the history force $\boldsymbol{F}_{\mathrm{h}} = \boldsymbol{F}_{\mathrm{Bass}}$ due to Basset [175]. However, depending on the specific problem, a wide variety of other physical phenomena might need to be included to obtain a reasonable approximation of the unresolved integral. For more in-depth discussions of these force terms we refer the reader to [26, 95, 108, 118]. An implementation of virtual mass and lift force in the context of CFD-DEM can be found, for example, in the solver by Sun et al. [150]. In most applications of fluid-solid flow the density ratio between fluid and solid will not be close to unity and in these cases, as noted in [118], the above effects are negligible compared to the effects of gravity and drag force. Thus, we do not consider these effects and leave their inclusion for future work, in order to concentrate on the dominant effect of drag within this thesis.

First, let us consider the drag force on a single isolated particle. Generally, this is expressed as (cf. [26, 118, 177])

$$\boldsymbol{F}_{\mathrm{d}} = \frac{1}{2} C_{\mathrm{d}} \rho_{\mathrm{f}} A_c \|\boldsymbol{u}_{\mathrm{f}} - \boldsymbol{v}_{\mathrm{s}}\| (\boldsymbol{u}_{\mathrm{f}} - \boldsymbol{v}_{\mathrm{s}}) \tag{3.4.2}$$

where C_{d} is the *drag coefficient* and A_c denotes the cross-sectional area of the particle with respect to the direction of relative velocity $\boldsymbol{u}_{\mathrm{f}} - \boldsymbol{v}_{\mathrm{s}}$. For the case of spherical particles, which we consider here, A_c describes the corresponding circle area such that

$$\boldsymbol{F}_{\mathrm{d}} = \frac{1}{8} C_{\mathrm{d}} \rho_{\mathrm{f}} \pi D_{\mathrm{s}}^2 \|\boldsymbol{u}_{\mathrm{f}} - \boldsymbol{v}_{\mathrm{s}}\| (\boldsymbol{u}_{\mathrm{f}} - \boldsymbol{v}_{\mathrm{s}}) \tag{3.4.3}$$

The drag coefficient C_{d} depends on a lot of different factors like the flow configuration, quantified, for example, in terms of Mach or Reynolds number, on the particles shape and

its orientation with respect to the relative flow direction. While we can circumvent the last dependency for spheres, it is well known that the coefficient can vary strongly with the single-phase particle Reynolds number

$$Re_\mathrm{p} = \frac{\rho_\mathrm{f} \|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\| D_\mathrm{s}}{\mu} \tag{3.4.4}$$

as is visualized for example in the text book by Crowe at al. [26, p.70]. In the regime of Stokes flow, $Re_\mathrm{p} < 1$, the coefficient is inversely proportional to the Reynolds number, i.e.

$$C_\mathrm{d} = \frac{24}{Re_\mathrm{p}} \tag{3.4.5}$$

but it tends towards a constant value in the range of $1 \ll Re_\mathrm{p} < 750$. In the literature on multiphase flow this transitional regime is modeled by a range of different correlations. A prominent extension of Stokes drag is commonly attributed to Schiller and Naumann [133] and given by

$$C_\mathrm{d} = \frac{24}{Re_\mathrm{p}}(1 + 0.15\, Re_\mathrm{p}^{0.687}) \tag{3.4.6}$$

which is applicable for Reynolds numbers lower than 800 [25, 26]. Note that the combined correlation

$$C_\mathrm{d} = \begin{cases} \dfrac{24}{Re_\mathrm{p}}(1 + 0.15\, Re_\mathrm{p}^{0.687}) & \text{if } Re_\mathrm{p} < 10^3 \\ 0.44 & \text{if } Re_\mathrm{p} > 10^3 \end{cases} \tag{3.4.7}$$

is also often attributed to Schiller and Naumann. This combined correlation can be considered as an attempt to fix the deficiency of (3.4.6), which has the erroneous limit of $C_\mathrm{d} = 0$ for $Re_\mathrm{p} \to \infty$. Instead, it accounts for the fact that the drag coefficient becomes almost constant with a variation of only about 13% around $C_\mathrm{d} = 0.445$ [25, 26] within the range of approximately $750 < Re_\mathrm{p} < 3.5 \times 10^5$. This range of Reynolds numbers is sometimes referred to as Newton regime (cf. [7]). Another correlation which is commonly used in place of the above expression by Schiller and Naumann, is

$$C_\mathrm{d} = \left(0.63 + \frac{4.8}{\sqrt{Re_\mathrm{p}}}\right)^2 \tag{3.4.8}$$

This expression is often attributed to Dallavalle [30], but according to [42], its true origin is apparently still unclear. This correlation does not suffer from the same defect of vanishing drag for large Reynolds numbers, instead it converges towards a value of $C_\mathrm{d} = 0.3969$ which is within the above-mentioned 13% range around $C_\mathrm{d} = 0.445$. As is to be expected, there is a wide variety of additional correlations describing the transitional and Newton regime, some of which can be found in [7, 25, 26]. Providing an extensive overview on this matter would certainly go beyond the scope of this thesis. We further limit ourselves to applications where $Re_\mathrm{p} \ll 1 \times 10^5$, such that we do not have to include the pressure drop due to boundary layers effects, which occur beyond the critical Reynolds number of about 3.5×10^5 [26].

So far we have only considered the drag onto a single (spherical) particle and observed that there are a variety of different expressions. This inevitably results from the fact that, even for this simple case, the derivation of analytical expressions can become very difficult. By extension, such exact formulations can not be expected to be available for the wide range of particle and flow configurations in CFD-DEM applications as well. Historically this problem has been addressed by estimating forces onto individual particles from empirically determinable quantities like pressure drop across a packed bed of particles or sedimentation

velocities [7, 36]. However, due to the rapid increase of computational resources within the last decades, numerous drag laws have been proposed on the basis of finely resolved numerical simulations instead of physical experiments. The amount of different correlations of either type provided in the literature on multiphase flow with dispersed phase and more specifically in unresolved CFD-DEM, prohibits a complete overview in the context of this thesis. Instead, we only provide a subset of some of the more commonly applied expressions to provide the reader with a general form and to simplify comparing our notations to those found in other publications. Of course, we also include any correlations that might not be "classical" ones, but will be used in the later verification stages.

Many of these drag laws, experimental as well as numerical, have in common that they can be classified in one of two groups identified by Beetstra et al. [7]. On the one hand, there are modifications of single particle drag laws due to the presence of additional particles and on the other hand there are expressions that extend formulas for particle configurations in Stokes flow to the inertial regime.

One of the most well known correlations is of the latter type, namely the pressure drop expression by Ergun [43]. It is based on the Forchheimer equation for porous media, which is itself an extension of the famous law by Darcy to the inertial regime. The general correlation proposed by Ergun is often written as

$$\frac{\Delta p}{L_\mathrm{b}} = A\frac{\mu}{D_\mathrm{s}^2}\frac{\varepsilon_\mathrm{s}^2}{\varepsilon_\mathrm{f}^3}u_\mathrm{sf} + B\frac{\rho_\mathrm{f}}{D_\mathrm{s}}\frac{\varepsilon_\mathrm{s}}{\varepsilon_\mathrm{f}^3}u_\mathrm{sf}^2 \tag{3.4.9}$$

where Δp denotes the pressure drop across a particle bed of length L_b in flow direction and u_sf denotes the so-called *superficial velocity*, also sometimes referred to as *volumetric flux*. The superficial velocity is the one that the fluid phase assumes if there are no particles present in the flow domain. If we consider a static block of particles, as is common for porous media, we realize that, since the mass flow needs to remain constant, the superficial velocity is related to the fluid-phase intrinsic average via

$$u_\mathrm{sf} = \varepsilon_\mathrm{f}\langle u_\mathrm{f}\rangle^\mathrm{f} \tag{3.4.10}$$

For moving dispersed phases it is common to use the correlations for static porous media and replace the above superficial velocity by the more general definition

$$\boldsymbol{u}_\mathrm{sf} = \varepsilon_\mathrm{f}\Big(\langle \boldsymbol{u}_\mathrm{f}\rangle^\mathrm{f} - \langle \boldsymbol{u}_\mathrm{s}\rangle^\mathrm{s}\Big) \tag{3.4.11}$$

which can be viewed as the difference between the average mixture velocity $\langle \boldsymbol{u}\rangle^\mathrm{t}$ and the solid phase intrinsic average velocity $\langle \boldsymbol{u}_\mathrm{s}\rangle^\mathrm{s}$ [118]. With this definition, the Ergun equation can be written in terms of the pressure gradient in the fluid phase as

$$-\boldsymbol{\nabla} p = A\frac{\mu}{D_\mathrm{s}^2}\frac{\varepsilon_\mathrm{s}^2}{\varepsilon_\mathrm{f}^3}\boldsymbol{u}_\mathrm{sf} + B\frac{\rho_\mathrm{f}}{D_\mathrm{s}}\frac{\varepsilon_\mathrm{s}}{\varepsilon_\mathrm{f}^3}\|\boldsymbol{u}_\mathrm{sf}\|\boldsymbol{u}_\mathrm{sf} \tag{3.4.12}$$

The parameters A, B in the above equation have been experimentally determined by Ergun from flows through randomly packed particles at volume fractions of $\varepsilon_\mathrm{f} \approx 0.4$. Although this correlation is commonly applied to monodisperse (all particles have the same shape and size) spherical particles, Beetstra et al. [7] pointed out that the original fitting has been partially obtained from experimental results on particles which were not exactly spherical. While the original fitting resulted in $A = 150$ and $B = 1.75$, other constant values [98] and a wide variety of non-constant modifications have been proposed since then. For an extensive overview we refer the reader to a review article by Yadzchi and Luding [169]. We further note, that for $A = 180$ and $B = 0$ the Ergun equation reduces to the so-called Kozeny-Carman equation.

So far we have only considered an expression for the pressure drop within the fluid phase, although we were originally interested in formulas for the drag force. One can derive a relation between these two quantities by assuming that the fluid flows with a spatially and temporally constant velocity through an assembly of particles with fixed porosity (cf. [7]). Then, the momentum equation in model II (3.3.3) reduces to

$$-\varepsilon_\mathrm{f}\boldsymbol{\nabla}p+\rho_\mathrm{f}\varepsilon_\mathrm{f}\boldsymbol{g}=\boldsymbol{f}_\mathrm{fs}^\mathrm{II} \tag{3.4.13}$$

If we further assume that the terms subsumed in $\boldsymbol{F}''$ are negligible and every particle experiences the same drag force, the coupling force density from (3.3.12) becomes

$$\boldsymbol{f}_\mathrm{fs}^\mathrm{II}(\boldsymbol{x})=\boldsymbol{F}_\mathrm{d}\sum_{i=1}^{N_\mathrm{s}}w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},i})=\boldsymbol{F}_\mathrm{d}n_\mathrm{p}(\boldsymbol{x}) \tag{3.4.14}$$

For particles of identical volume we further saw that $\varepsilon_\mathrm{s}\approx n_\mathrm{p}V_\mathrm{s}$, such that we finally reformulate (3.4.13) to obtain

$$\boldsymbol{F}_\mathrm{d}=-\frac{\varepsilon_\mathrm{f}V_\mathrm{s}}{\varepsilon_\mathrm{s}}(\boldsymbol{\nabla}p-\rho_\mathrm{f}\boldsymbol{g}) \tag{3.4.15}$$

which coincides with the relation found in [125]. As is also mentioned by the authors in this article, the gravity term is often neglected such that (see also [7])

$$\boldsymbol{F}_\mathrm{d}=-\frac{\varepsilon_\mathrm{f}V_\mathrm{s}}{\varepsilon_\mathrm{s}}\boldsymbol{\nabla}p \tag{3.4.16}$$

which is due to the fact that these relations are often derived for comparatively low fluid densities, where the gravity term is in fact very small compared to the pressure drop.

If we consider the second one of the above-mentioned groups of correlations, i.e. modifications of single particle drag laws due to the presence of other particles, they are typically written as

$$\boldsymbol{F}_\mathrm{d}=\frac{1}{8}C_\mathrm{d}\rho_\mathrm{f}\pi D_\mathrm{s}^2\|\boldsymbol{u}_\mathrm{sf}\|\boldsymbol{u}_\mathrm{sf}g(\varepsilon_\mathrm{f}) \tag{3.4.17}$$

in our case of spherical particles. When comparing this formulation with (3.4.3), we observe that the relative velocity is replaced by the superficial velocity and that the so-called *voidage function* g has been introduced to account for the presence of other particles. We note that the particle Reynolds number, used within the drag coefficient, is now also dependent on the superficial velocity, such that

$$Re_\mathrm{p}=\frac{\varepsilon_\mathrm{f}\rho_\mathrm{f}\|\boldsymbol{u}_\mathrm{f}-\boldsymbol{v}_\mathrm{s}\|D_\mathrm{s}}{\mu} \tag{3.4.18}$$

Unless stated otherwise, we usually refer to this definition when talking about the Reynolds number in the following. For the most important part of the above modification of single particle drag, i.e. the voidage function, it is very common to assume the form

$$g(\varepsilon_\mathrm{f})=\varepsilon_\mathrm{f}^{-\chi} \tag{3.4.19}$$

where the exponent χ was originally chosen to be a constant value. An example for this is the well-kown correlation by Wen and Yu [163], which results from the extended Schiller Naumann drag coefficient (3.4.7) and $\chi=3.7$. Since the initial work by Wen and Yu, several authors proposed extensions to non-constant exponents. Di Felice [35] considered a drag law of the form (3.4.17) with drag coefficient (3.4.8) and concluded from available experimental

results, that the exponent χ is dependent on the particle Reynolds number. From a fitting to this data, they obtained

$$\chi = 3.7 - 0.65 \exp\left(-\frac{(1.5 - \log_{10}(Re_\mathrm{p}))^2}{2}\right) \tag{3.4.20}$$

which clearly is a modification of the constant value $\chi = 3.7$ by Wen and Yu. Different combinations of this exponent and drag coefficients occur in the literature. For example Zhu et al. [177] use the extended Schiller Naumann coefficient (3.4.7) while Epstein [42] proposed the usage of

$$C_\mathrm{d} = \left(0.54 + \frac{4.8}{\sqrt{Re_\mathrm{p}}}\right)^2 \tag{3.4.21}$$

instead of (3.4.8) to increase the range of validity in terms of particle Reynolds number up to $Re_\mathrm{p} \leq 3000$. According to Epstein, Di Felice's original result is only applicable for $\varepsilon_\mathrm{f} \geq 0.4$, i.e. the solid particles are not allowed to be too densely packed. This goes in line with the common observation that single particle drag modifications are more suitable for dilute flows. A different extension which also accounts for dependencies on the Reynolds number was, for example, proposed by Rong et al. [125] on the basis of lattice Boltzmann simulations. These modifications aim to resolve the limitation of the original correlation to rather dilute flow and provide single expressions for a larger range of flow configurations. Alternatively, combinations of correlations obtained from dense and dilute particulate phases have been proposed. The commonly used [115, 162, 173] combination of the pressure drop equation by Ergun and a drag law similar to the one by Wen and Yu is often attributed to Gidaspow [55]. Since in [55], the author covers two-fluid models, the expression is not formulated in terms of drag force on individual particles, but via the so-called *inter-phase momentum transfer coefficient* or *fluid friction coefficient* (cf. [108]) β. However, they are related via (cf. [34, 108, 168])

$$\boldsymbol{F}_\mathrm{d} = \frac{V_\mathrm{s}\beta}{\varepsilon_\mathrm{s}}(\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}) \tag{3.4.22}$$

which can also be seen from the fact that Gidaspow used the relation

$$-\varepsilon_\mathrm{f}\boldsymbol{\nabla} p - \beta_A(\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}) = 0 \tag{3.4.23}$$

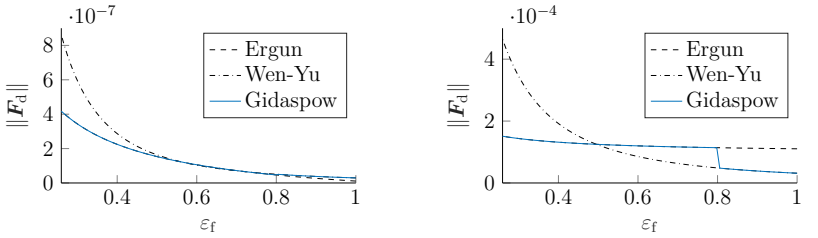
to obtain β and from the connection between pressure drop and drag force in (3.4.16). From this it is also clear that β is typically a function of fluid volume fraction and relative velocity of fluid and solid phase, i.e.

$$\beta = \beta(\varepsilon_\mathrm{f}, \|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\|) \tag{3.4.24}$$

and that β in our notation corresponds to friction coefficient β_A in model A by Gidaspow. The combined drag law then takes the form

$$\beta = \begin{cases} 150\dfrac{\varepsilon_\mathrm{s}^2\mu}{\varepsilon_\mathrm{f} D_\mathrm{s}^2} + 1.75\dfrac{\rho_\mathrm{f}\|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\|\varepsilon_\mathrm{s}}{D_\mathrm{s}} & \text{if } \varepsilon_\mathrm{f} \leq 0.8 \\ \dfrac{3}{4}C_\mathrm{d}\dfrac{\rho_\mathrm{f}\varepsilon_\mathrm{s}\varepsilon_\mathrm{f}}{D_\mathrm{s}}\|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\|\varepsilon_\mathrm{f}^{-2.65} & \text{if } \varepsilon_\mathrm{f} > 0.8 \end{cases} \tag{3.4.25}$$

where C_d is given by the extended Schiller Naumann coefficient (3.4.7) and the exponent has been altered slightly from $\chi = 3.7$, as originally proposed by Wen and Yu, to $\chi = 3.65$ as in [56]. It is important to note that this formulation has a discontinuity at $\varepsilon_\mathrm{f} = 0.8$



(a) Relative velocity of $\|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\| = 0.1\,\mathrm{m/s}$ **(b)** Relative velocity of $\|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\| = 10\,\mathrm{m/s}$

Figure 3.4: Plots of the combined drag law (3.4.25) proposed by Gidaspow and the individual drag laws by Ergun as well as Wen and Yu. The expressions are evaluated for a particle with $D_\mathrm{s} = 1\,\mathrm{mm}$ in air ($\rho_\mathrm{f} = 1.205\,\mathrm{kg/m^3}$, $\mu = 1.813 \times 10^{-5}\,\mathrm{Pa\,s}$). Two fixed relative velocities are considered and the magnitude of drag force is plotted over a range of fluid volume fractions.

with a jump that increases with relative velocity between fluid and solid phase. This can be seen in Figure 3.4, where we plotted the individual drag laws as well as their combination by Gidaspow over a range of fluid volume fractions at two different relative velocities. The fluid volume fractions are chosen so that they represent the range of physically reasonable values for a two-phase flow with monodisperse spherical particles as the solid phase. Clearly, the continuous fluid phase is able to fill out any volume within the computational domain which is not occupied by such particles. In particular, regions with no solid particles can exist and lead to a porosity of $\varepsilon_\mathrm{f} = 1$, which is the trivial upper bound for this quantity. On the other hand, thinking for example of a cannonball stack, such a configuration of equally sized spheres will have voids in between the balls. These voids mean that $\varepsilon_\mathrm{s} < 1$ and in our two-phase setting consequently $\varepsilon_\mathrm{f} > 0$. This already suggests what will be discussed in more detail in Appendix B, namely that there is a lower bound for the porosity in the setting of Figure 3.4.

Returning to the analysis of (3.4.25), we also want to note that in terms of volume fraction, the Ergun correlation has been extended far beyond its original experimental setup. Thus, despite its popularity in CFD-DEM literature, one has to be mindful about possible instabilities or wrong drag predictions in the intermediate regime of volume fraction. As noted in [34], this can, to some extent, be alleviated by taking the minimum of both correlations instead of choosing a threshold value. But the authors further conclude that, even then, more recent drag correlations such as the one by Hill et. al. [61, 78] and Beetstra et al. [7] have shown better agreement with experiments performed by Link et. al. [92] in the context of gas-solid fluidized beds.

At this point we want to stop our discussion on drag correlations, since an extensive review with comments on all the drag expression which are commonly used in unresolved CFD-DEM would certainly be beyond the scope of this thesis. Instead, we provide tabular overviews (see Table 3.1 and Table 3.2) on a small subset of important expressions, including in particular those which have already been discussed and will be used in the numerical scheme. Comments on the latter ones are postponed to the sections in which they are applied. Note that the tables have been split into correlations formulated in the voidage function exponent χ and in the transfer coefficient β for better readability. For a short analysis on the correlations of the former type we refer to Appendix C.

When searching for further correlations in CFD-DEM literature, we urge the reader to pay close attention to the force which is described by the proposed expression. This is important since the experimental or simulation setup which leads to such a correlation can provide the

total fluid-solid interaction force $\boldsymbol{F}_{\mathrm{fs}}$ instead of $\boldsymbol{F}_{\mathrm{d}}$. For monodisperse particle assemblies, under the specific flow conditions of these experiments, one can reformulate these expressions via [108]

$$\boldsymbol{F}_{\mathrm{d}} = \varepsilon_{\mathrm{f}} \boldsymbol{F}_{\mathrm{fs}} \tag{3.4.26}$$

On the other hand, for polydisperse systems this does not hold true, which has led to some mistakes in literature [44, 130]. In modeling these systems, specialized drag closures are necessary to represent phenomena such as layer inversion [37]. To cross out another one of the numerous pitfalls connected to drag laws, we want to highlight the fact that it is not clear a priori whether the above correlations can be applied in conjunction with volume fractions from two-dimensional simulations. This is discussed, for example, in [108], where the authors provide a conversion formula to obtain an equivalent volume fraction which can be used to evaluate the drag law. While this is not of importance in the context of this work, since we only consider three dimensional settings, it further emphasizes the amount of thought which needs to be put into the development of a scheme which realizes the discussed unresolved two-way coupling. Accordingly, we devote the upcoming chapter solely to this task.

Original work	Method	Voidage function exponent
Wen&Yu (1966) [163]	Exp.	$\chi = 3.7$
Di Felice (1994) [35]	Exp.	$\chi = 3.7 - 0.65 \exp\left[-\frac{(1.5-\log_{10}(Re_\mathrm{p}))^2}{2}\right]$
Rong et al. (2013) [125]	LB	$\chi = 2.65(\varepsilon_\mathrm{f}+1) - (5.3-3.5\varepsilon_\mathrm{f})\varepsilon_\mathrm{f}^2 \exp\left[-\frac{(1.5-\log_{10}(Re_\mathrm{p}))^2}{2}\right]$

Table 3.1: Overview of some drag force correlations applicable to spherical monodisperse particles and formulated in terms of voidage function exponent χ. The "Method" column shows whether the expression has been derived based on experimental data (Exp.) or from resolved Lattice-Boltzmann simulations (LB). For a given form of χ, the drag force is obtained from (3.4.17) and (3.4.19). The drag coefficient C_d is typically chosen as in (3.4.8)

Original work	Method	Inter-phase momentum transfer coefficient
Ergun (1952) [43]	Exp.	$\beta = 150\frac{\varepsilon_\mathrm{s}^2\mu}{\varepsilon_\mathrm{f}(D_\mathrm{s})^2} + 1.75\frac{\rho_\mathrm{f}\|\boldsymbol{u}_\mathrm{f}-\boldsymbol{v}_\mathrm{s}\|\varepsilon_\mathrm{s}}{D_\mathrm{s}}$
Koch&Hill (2001) [78]	LB	$\beta = \frac{18\mu\varepsilon_\mathrm{f}^2\varepsilon_\mathrm{s}}{D_\mathrm{s}^2}\left[A + \frac{1}{2}B\,Re_\mathrm{p}\right]$ $A = \begin{cases} \frac{1+3\sqrt{\frac{\varepsilon_\mathrm{s}}{2}}+\frac{135}{64}\varepsilon_\mathrm{s}\ln(\varepsilon_\mathrm{s})+16.14\varepsilon_\mathrm{s}}{1+0.681\varepsilon_\mathrm{s}-8.48\varepsilon_\mathrm{s}^2+8.16\varepsilon_\mathrm{s}^3} & \text{if } \varepsilon_\mathrm{s} < 0.4 \\ \frac{10\varepsilon_\mathrm{s}}{\varepsilon_\mathrm{f}^3} & \text{if } \varepsilon_\mathrm{s} \geq 0.4 \end{cases}$ $B = 0.0673 + 0.212\varepsilon_\mathrm{s} + \frac{0.0232}{\varepsilon_\mathrm{f}^5}$
Beetstra et al. (2007) [7]	LB	$\beta = \frac{18\mu\varepsilon_\mathrm{f}^2\varepsilon_\mathrm{s}}{D_\mathrm{s}^2}\left[A + \frac{1}{2}B\,Re_\mathrm{p}\right]$ $A = \frac{10\varepsilon_\mathrm{s}}{\varepsilon_\mathrm{f}^3} + \varepsilon_\mathrm{f}(1+1.5\sqrt{\varepsilon_\mathrm{s}})$ $B = \frac{0.413}{12\varepsilon_\mathrm{f}^3}\left[\frac{\frac{1}{\varepsilon_\mathrm{f}}+3\varepsilon_\mathrm{s}\varepsilon_\mathrm{f}+8.4\,Re_\mathrm{p}^{-0.343}}{1+10^{3\varepsilon_\mathrm{s}}\,Re_\mathrm{p}^{-(1+4\varepsilon_\mathrm{s})/2}}\right]$

Table 3.2: Overview of some drag force correlations applicable to spherical monodisperse particles and formulated in terms of momentum transfer coefficient β. The "Method" column shows whether the expression has been derived based on experimental data (Exp.) or from resolved Lattice-Boltzmann simulations (LB). For a given form of β, the drag force is obtained from (3.4.22).

Chapter 4

CFD-DEM in a GFDM framework

In the previous chapter we have derived a set of volume averaged equations that describe the fluid phase behavior on a scale which is significantly larger than the individual solid particle size. This model is valid in the whole domain covered by any of the phases, meaning that we consider inter-penetrating continua. While similar procedures lead to continuous descriptions in both phases, we also discussed our choice of using a numerical method to resolve individual particle movement. The phase interaction is, on the one hand, given by coupling forces such as drag, which represent the unresolved effects on particle surfaces due to the surrounding fluid. On the other hand, the fluid phase is influenced by the presence of solid particles through the volume fraction, which creates regions of reduced effective material density and through momentum source terms, which are calculated according to Newton's third law from coupling forces acting on the particles. From these considerations we can immediately identify three major components of a CFD-DEM scheme, namely the solver for solid phase motion, the discrete element method, a numerical scheme for calculating the volume fraction and other averages, often referred to as coarse graining in the context of CFD-DEM and finally the numerical scheme for solving the volume averaged Navier-Stokes (VANS) equations. We will embed these components in a time integration loop, as sketched below in Figure 4.1, which forms the complete CFD-DEM scheme.

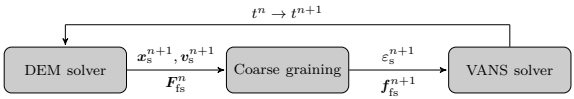


Figure 4.1: Rough outline of the main loop within the CFD-DEM algorithm presented in this work.

The reason for this order of computation is the fact that instead of the incompressibility constraint we have a dependency of velocity divergence on the change in volume fraction (cf. (3.2.95)). If the particles are moved before the VANS solver, one can use their new positions to calculate a backward difference approximation to this temporal change and impose a condition for the velocity divergence at the new time level. More details on this subject-matter will be given when we consider the numerical scheme for the fluid phase at a later point within this chapter.

This chapter is structured in the same way that is visualized in the above sketch of our coupled scheme. First, we shed some light on the methodology behind DEM and describe some numerical aspects within our framework. Next, we analyze how discrete solid phase averages are calculated in the FPM to provide the volume fraction and source terms described analytically in the previous chapter. Finally, we introduce the numerical scheme that is used to solve the volume averaged equations for the fluid phase and discuss different time step restrictions in the complete coupled solver.

4.1 Discrete element method

In Section 3.1 we already described the discrete element method as an approach which typically considers every solid entity in the physical system as a discrete point in the numerical scheme and resolves collisions between individual particles to obtain the bulk behavior. This reveals a tight connection to molecular dynamics simulations from which the method originated. We note that the term DEM has been ambiguously used in literature for two different methods of resolving collisions between particles, called *soft-sphere* and *hard-sphere* DEM.

In the latter approach, described for example in [34, 66, 97, 161], particles are assumed to be perfectly rigid and their collisions binary (collisions between exactly two particles) and instantaneous. Contacts are resolved within a single time step by prescribing the relation between pre-collisional and post-collisional velocities on the basis of momentum conservation. This method can be very efficient when collisions occur infrequently, since in this case it is typically incorporated into so-called event-driven algorithms which are able to integrate the particle trajectory with large time steps in between successive collisions. The major downside of this approach is the fact that it is not able to model dense configurations of particles since they violate the assumptions of binary and instantaneous collisions. We will see that the more prominent soft-sphere approach does not have this problem, albeit at the cost of increased computational cost.

As we already remarked in the previous chapter, the method we want to employ is characterized by allowing particles to have an overlap in the numerical scheme. Based on this overlap, forces are calculated that recover the non-overlapping state and aim to represent the experimentally observable rebound behavior. This version of DEM is the one typically referred to as soft-sphere method. The name originates from the concept that overlaps represent deformations of particles due to collisions. However, it is important to realize that for solid particles the physically occurring deformations are very small and the overlap within a simulation is more of a modeling tool than a representation of these deformations. Clearly, to ensure that this simplification does not misrepresent the true behavior of a granular assembly, tight bounds on the overlap size and strong repulsive forces are necessary. In contrast to the hard-sphere approach, particles have to stay in contact for several time steps in order to ensure an accurate integration of these forces throughout the different phases of the collision. As we will later see, the strength of these repulsive forces imposes severe time restrictions on the numerical scheme. We will discuss these restrictions at the end of this section when we have discussed a few typical contact models. We also note that it has been common practice to allow for nonphysically large overlaps in order to relax these bounds on the time step.

4.1.1 Equations of motion

Independent of the individual particle shape, a system of N_s interacting particles with masses $m_{\mathrm{s},i}$ obeys the Newton equations describing the change in positions $\boldsymbol{x}_{\mathrm{s},i}$ and velocities $\boldsymbol{v}_{\mathrm{s},i}$ with respect to time

$$\frac{\mathrm{d}\boldsymbol{x}_{\mathrm{s},i}}{\mathrm{d}t} = \boldsymbol{v}_{\mathrm{s},i} \tag{4.1.1}$$

$$m_{\mathrm{s},i}\frac{\mathrm{d}\boldsymbol{v}_{\mathrm{s},i}}{\mathrm{d}t} = \boldsymbol{F}_{\mathrm{b},i} + \sum_{\substack{j=1 \\ j\neq i}}^{N_\mathrm{s}} \boldsymbol{F}_{\mathrm{c},ij} \tag{4.1.2}$$

Here we denote by $\boldsymbol{F}_{\mathrm{b},i}$ all the body forces acting on particle i while $\boldsymbol{F}_{\mathrm{c},ij}$ denotes the contact forces due to a collision between particle i and j.

For the sake of completeness, we note that, if rotations are considered, one would additionally have to solve the equations

$$\frac{\mathrm{d}\boldsymbol{\theta}_{\mathrm{s},i}}{\mathrm{d}t} = \mathbf{S}(\boldsymbol{\theta}_{\mathrm{s},i})\boldsymbol{\omega}_{\mathrm{s},i} \tag{4.1.3}$$

$$\mathrm{I}_{\mathrm{s},i}\frac{\mathrm{d}\boldsymbol{\omega}_{\mathrm{s},i}}{\mathrm{d}t} = \boldsymbol{T}_{\mathrm{b},i} + \sum_{\substack{j=1 \\ j\neq i}}^{N_\mathrm{s}} \boldsymbol{T}_{\mathrm{c},ij} \tag{4.1.4}$$

for spherical particles with scalar moment of inertia $\mathrm{I}_{\mathrm{s},i}$. The components of $\boldsymbol{\theta}_{\mathrm{s},i}$ are variables from the parametrization of the rotation matrix space $SO(3)$ and describe the orientation of a coordinate system attached to the rotating body relative to a global coordinate system. The form of $S(\boldsymbol{\theta}_{\mathrm{s},i})$ is defined by the specific choice of parametrization. In the second equation, the variable $\boldsymbol{\omega}_{\mathrm{s},i}$ denotes the angular velocity in the global coordinate system while $\boldsymbol{T}_{\mathrm{c},ij}$, $\boldsymbol{T}_{\mathrm{b},i}$ again denote torques induced by particle interactions of particle i with particle j and other torques applied directly to the body, respectively. A more precise explanation on the subject-matter of different coordinate systems and rotational motion is provided in Appendix D. In order to focus on the coupling and since rotations are not mandatory for the coupled numerical simulations presented in Chapter 5, we have not considered them in the context of this work. Still, we provide points of contact with this topic for both, the interested reader and for later extensions of our work, by including the rotational velocities when we consider frictional contributions to the contact forces.

Within the generalized finite difference framework introduced in Chapter 2, each solid particle is represented by a single point. Thus, we keep track of (at least) two point clouds, one set of points with positions

$$\boldsymbol{x}_{\mathrm{f},i}, \quad i = 1, \ldots, N_\mathrm{f}$$

that represent the fluid phase discretization and one with positions

$$\boldsymbol{x}_{\mathrm{s},i}, \quad i = 1, \ldots, N_\mathrm{s}$$

that represent the actual locations of solid particle centroids. Since points within the solid point cloud correspond to physical entities rather than a numerical discretization, we obviously do not apply any of the measures discussed in Chapter 2 that would lead to point addition or removal outside of inflow or outflow boundaries. We will further discuss the movement of solid points, i.e. the integration scheme for translational motion, separately from the discretization points to tailor towards the needs of the discrete element method. As we have described in Chapter 2, the construction of differential operators for a given point requires the identification of all points of the same phase which are within a distance determined by the smoothing length h. The attentive reader might already realize that we are faced with a very similar task in DEM. In order to check if a collision between two spherical particles occurs, we need to check if the distance between their centroids is larger than the sum of their radii. We can thus specify a smoothing length equal to the maximum diameter among particles in the granular assembly and use the same exact algorithms of neighbor search as for the continuous phase. We note that the same would be possible for aspherical particles, by instead using the diameter of circumscribed spheres. This effect provides a very useful synergy, since improvements in the neighbor scheme, which is crucial for the performance of a discrete element method, can also improve the continuous phase computations.

4.1.2 Normal contact models

As stated above, the general concept of a soft-sphere discrete element method is to allow numerically represented particles to penetrate each other and calculate forces from a mea-

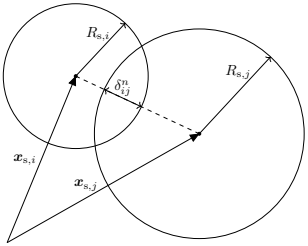


Figure 4.2: Visualization of normal overlap in the collision of two spherical particles.

sure of the created overlap as well as relative velocities. The most obvious measure is the geometrical volume of the region occupied by both particles. To define a different measure, we note that two colliding convex particles in $\mathbb{R}^d$ create a bounded overlap region in $\mathbb{R}^{d-1}$. Taking the normal on these regions in $\mathbb{R}^{d-1}$, the overlap is then quantified as the maximum extent along this direction. It is not a priori clear which measure of overlap size to choose and it certainly depends on the particle shape as well as the contact model used to calculate the forces. For spherical particles, however, the latter definition is almost exclusively used. We realize that (see Figure 4.2) for two colliding spherical particles with positions $\boldsymbol{x}_{s,i}$, $\boldsymbol{x}_{s,j}$ and radii $R_{s,i}$, $R_{s,j}$ the *normal overlap* can be simply written as

$$\delta_{ij}^n = R_{s,i} + R_{s,j} - \|\boldsymbol{x}_{s,i} - \boldsymbol{x}_{s,j}\| \tag{4.1.5}$$

Furthermore, we define the *contact normal*

$$\boldsymbol{n}_{ij} = \frac{\boldsymbol{x}_{s,i} - \boldsymbol{x}_{s,j}}{\|\boldsymbol{x}_{s,i} - \boldsymbol{x}_{s,j}\|} \tag{4.1.6}$$

which is orthogonal to the overlap region in $\mathbb{R}^{d-1}$ and oriented towards particle i. With these definitions a purely elastic collision between particles i and j is typically modeled by applying a single force

$$\boldsymbol{F}_{c,ij}^n = \begin{cases} F_{el}\left(\delta_{ij}^n\right)\boldsymbol{n}_{ij} & \text{if } \delta_{ij}^n \geq 0 \\ 0 & \text{else} \end{cases} \tag{4.1.7}$$

to particle i and its opposing force

$$\boldsymbol{F}_{c,ji}^n = -\boldsymbol{F}_{c,ji}^n \tag{4.1.8}$$

to particle j. We note that, for spherical particles, the normal forces only influence the translational motion since they are always applied to a material point along the line connecting the sphere centroids. We will further discuss this at a later point and assume for now that $\boldsymbol{F}_{c,ij}^n$ is applied to the centroid of particle i. When we consider extensions of the above purely elastic force, we also need a notion of normal velocity. To this end we denote the relative translational velocity of centroids by

$$\boldsymbol{v}_{s,ij} = \boldsymbol{v}_{s,i} - \boldsymbol{v}_{s,j} \tag{4.1.9}$$

and define the *relative normal velocity*

$$v_{s,ij}^n = \boldsymbol{v}_{s,ij} \cdot \boldsymbol{n}_{ij} \tag{4.1.10}$$

With this definition it is customary to introduce a dissipative or damping term in addition to the elastic force in (4.1.7) to obtain

$$\boldsymbol{F}_{c,ij}^n = \left[F_{el}\left(\delta_{ij}^n\right) + F_{diss}\left(\delta_{ij}^n, v_{ij}^n\right)\right]\boldsymbol{n}_{ij} \tag{4.1.11}$$

Note that, in order to improve readability here and in the following, we do not explicitly state that forces are zero for negative overlap. Now that we have established the general notation, we want to provide details on the form of F_{el} and F_{diss}. The particular expressions for these forces, based on overlap and also relative velocities, are commonly referred to as *contact models* in DEM.

Fixed stiffness spring models

The origins of the discrete element method are typically traced back to Cundall and Strack [29], who proposed the usage of a force-displacement law based on springs to model the behavior of two-dimensional assemblies of discs. In the above notation this leads to a normal force of the form

$$\boldsymbol{F}_{\text{c},ij}^{n} = k_{\text{n}}^{*}\delta_{ij}^{n}\boldsymbol{n}_{ij} \tag{4.1.12}$$

where k_{n}^{*} is the (constant) normal spring stiffness and δ_{ij}^{n} is the normal overlap at the contact point. We reiterate that the sole purpose of these springs is to prevent unphysical penetration of particles in the computer simulation of the assembly and that there is no inherent connection to material properties at this point. Therefore the springs in a constant spring stiffness model should rather be viewed as "penalty springs" [109]. Regardless, we need some criterion to choose reasonable values for the coefficient k_{n}^{*}. A rather simplistic approach to this problem is presented in [99] where the authors propose the criterion

$$k_{\text{n}}^{*} \geq \frac{m_{\text{s}}^{*}\left(v_{\text{s},ij}^{n,0}\right)^{2}}{\left(\delta_{ij}^{\text{n,max}}\right)^{2}} \tag{4.1.13}$$

for a relative normal velocity $v_{\text{s},ij}^{n,0}$ before the collision, a maximum allowed overlap $\delta_{ij}^{\text{n,max}}$ and the effective particle mass in a binary interaction

$$\frac{1}{m_{\text{s}}^{*}} = \frac{1}{m_{\text{s},i}} + \frac{1}{m_{\text{s},j}} \quad \Leftrightarrow \quad m_{\text{s}}^{*} = \frac{m_{\text{s},i}m_{\text{s},j}}{m_{\text{s},i} + m_{\text{s},j}} \tag{4.1.14}$$

We make use of the effective mass at several points within this chapter and want to note that for the contact of a particle i with a (rigid and immovable) wall one recovers $m_{\text{s}}^{*} = m_{\text{s},i}$ by assigning infinite mass to the wall. The formula (4.1.13) is easily obtained by equating the kinetic energy before collision and potential energy due to the springs at maximum deformation. As noted before, the maximum overlap in DEM is typically required to be a very small fraction of the particle radius and can be specified a priori as part of the modeling, while the collision velocities within the system might not be known in advance. A rough guess for this quantity is proposed by the authors [99] as the velocity that a particle obtains in free fall along the longest path in gravitational direction within the considered system. Let us assume that the length of this path is given by h_0, then one obtains

$$v_{\text{s},ij}^{n,0} \geq \sqrt{2gh_0} \tag{4.1.15}$$

and

$$k_{\text{n}}^{*} = \frac{2m_{\text{s}}^{*}gh_0}{\left(\delta_{ij}^{\text{n,max}}\right)^{2}} \tag{4.1.16}$$

Again, this can be obtained by a conservation of energy argument from the potential energies of the gravitational field and kinetic energy before impact.

We note that in the case of granular assemblies with different types of particles it might be desirable to assign different penalties, i.e. have spring stiffnesses $k_{\text{n},i}$ assigned individually

to particles $i = 1, \ldots, N_\mathrm{s}$. Then, for two colliding particles let the deformation of the corresponding springs be denoted by δ_i, δ_j. Since the total deformation needs to be identical to the overlap and the forces onto the particles only differ in direction, we obtain

$$\delta_{ij}^n = \delta_i + \delta_j \tag{4.1.17}$$

$$k_{\mathrm{n},i}\delta_i = k_{\mathrm{n},j}\delta_j \tag{4.1.18}$$

If we solve this for a given overlap this leads to the individual deformations

$$\delta_i = \frac{k_{\mathrm{n},j}}{k_{\mathrm{n},i} + k_{\mathrm{n},j}}\delta_{ij}^n, \qquad \delta_j = \frac{k_{\mathrm{n},i}}{k_{\mathrm{n},i} + k_{\mathrm{n},j}}\delta_{ij}^n \tag{4.1.19}$$

and we can then calculate the spring stiffness in terms of total overlap as

$$k_\mathrm{n}^* = \frac{k_{\mathrm{n},i}k_{\mathrm{n},j}}{k_{\mathrm{n},i} + k_{\mathrm{n},j}} \quad \Leftrightarrow \quad \frac{1}{k_\mathrm{n}^*} = \frac{1}{k_{\mathrm{n},i}} + \frac{1}{k_{\mathrm{n},j}} \tag{4.1.20}$$

We note that this definition also reduces to the expected value of $k_{\mathrm{n},i}$ when body j is considered completely rigid, i.e. $k_{\mathrm{n},j} \to \infty$. Thus, when resolving the collision between two particles with different spring stiffnesses, one uses (4.1.12) with spring stiffness (4.1.20) to calculate the force onto particle i, and the force onto particle j is given by the force of same magnitude but reverse direction.

Elasticity theory and Hertzian contact

A model which addresses the missing relation of contact model parameter and material properties is the *Hertz model*. The corresponding Hertzian contact theory provides a rigorous derivation of pressure distribution and force in the non-adhesive contact of two elastic spheres based on the theory of elasticity. Details on this derivation can for example be found in the book by Popov [116], although a short discussion can also be found in the book on DEM by O'Sullivan [109]. Here, we mostly restrict ourselves to those parts which are relevant within our context. Following the Hertzian theory, the authors in [116] show that the deformation in a binary collision of non-adhesive elastic spheres is equivalent to that caused by a point force of magnitude

$$F = \frac{4}{3}\mathrm{Y}^*\sqrt{R_\mathrm{s}^*}\delta^{\frac{3}{2}} \tag{4.1.21}$$

with R_s^* denoting the effective or equivalent particle radius

$$\frac{1}{R_\mathrm{s}^*} = \frac{1}{R_{\mathrm{s},i}} + \frac{1}{R_{\mathrm{s},j}} \quad \Leftrightarrow \quad R_\mathrm{s}^* = \frac{R_{\mathrm{s},i}R_{\mathrm{s},j}}{R_{\mathrm{s},i} + R_{\mathrm{s},j}} \tag{4.1.22}$$

The symbol Y^* further denotes the effective *Young modulus* calculated from the moduli of individual particles via

$$\frac{1}{\mathrm{Y}^*} = \frac{1-\nu_i^2}{\mathrm{Y}_i} + \frac{1-\nu_j^2}{\mathrm{Y}_j} \tag{4.1.23}$$

while ν denotes the *Poisson ratio*. Both of the latter quantities are material properties, Young's modulus being the proportionality coefficient in the linear regime of the stress-strain relationship (at near-zero stress and strain) and Poisson's ratio characterizing the relation between axial elongation and contraction of cross-sectional area within a material. Again, if the particle j is assumed to be rigid, e.g. when we consider a wall collision, the effective properties instead become

$$R_\mathrm{s}^* = R_{\mathrm{s},i}, \qquad \mathrm{Y}^* = \frac{\mathrm{Y}_i}{1-\nu_i^2} \tag{4.1.24}$$

We note that the corresponding contact model can now be written as

$$\boldsymbol{F}_{\mathrm{c},ij}^{n} = k_{\mathrm{n}}^{\mathrm{H}}\left(\delta_{ij}^{n}\right)\delta_{ij}^{n}\boldsymbol{n}_{ij} \tag{4.1.25}$$

with a varying spring stiffness

$$k_{\mathrm{n}}^{\mathrm{H}}(\delta) = \tilde{k}_{\mathrm{n}}^{\mathrm{H}}\sqrt{\delta} \tag{4.1.26}$$

$$\tilde{k}_{\mathrm{n}}^{\mathrm{H}} = \frac{4}{3}\mathrm{Y}^{*}\sqrt{R_{\mathrm{s}}^{*}} \tag{4.1.27}$$

Although there is a theoretical derivation of these expressions from elasticity theory, it is important to realize the simplifications that were made in its course. A detailed discussion on this can be found in [109]. Here we only want to point out two assumptions. Namely, that not every material and deformation follows the presupposed linear elastic behavior and that the theory assumes frictionless particles which have no surface asperities, i.e. they are perfectly smooth spheres. These points already make it abundantly clear that not every physical object or every granular flow configuration is described precisely by the above contact forces. For example sand grains do typically strongly differ in shape and have nonsmooth surfaces. Nevertheless, contact models based on the Hertzian theory are well-established, frequently used and have shown great success in the context of the discrete element method. This is also emphasized by its wide adoption in both, commercial codes such as EDEM[1] and PFC[2], as well as non-commercial ones such as Yade[3] and LIGGGHTS[4]. The principal advantage of this model is the fact that there is a clear connection between contact model parameters and material properties. We do however want to note that with the same argument of energy conservation as before (see (4.1.13)) one can derive the expression

$$\tilde{k}_{\mathrm{n}}^{\mathrm{H}} \geq \frac{5}{4}\frac{m_{\mathrm{s}}^{*}\left(v_{\mathrm{s},ij}^{n,0}\right)^{2}}{\left(\delta_{ij}^{\mathrm{n,max}}\right)^{5/2}} \tag{4.1.28}$$

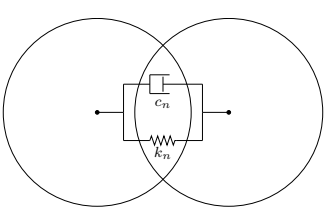
which can be used to prescribe a value of $\tilde{k}_{\mathrm{n}}^{\mathrm{H}}$ based on a maximum overlap and collision velocity instead of calculating it from material properties. Additionally, by reformulating the above equation in terms of velocity, replacing $\tilde{k}_{\mathrm{n}}^{\mathrm{H}}$ by the value from (4.1.27) and setting $\delta_{ij}^{\mathrm{n,max}} = R_{\mathrm{s},i}+R_{\mathrm{s},j}$, one can determine the maximum collision velocity which can be supported by the contact model. Clearly, for such velocities the Hertzian model no longer represents a realistic elastic behavior. We note that one can analogously calculate the maximum collision velocity supported by a linear spring by reformulating (4.1.13) in terms of velocity (see also Appendix E).

Linear spring-dashpot model

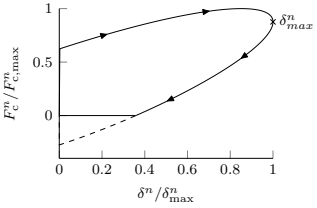
So far we have only described purely elastic contact models. However, in a real granular system kinetic energy is dissipated during collisions due to plastic deformations. A straightforward approach to model this phenomenon in the context of DEM is the addition of a damping term to the purely elastic contact model. For the linear spring model (4.1.12) the most prominent extension is of the form

$$\boldsymbol{F}_{\mathrm{c},ij}^{n} = \left[k_{\mathrm{n}}^{*}\delta_{ij}^{n} - d_{\mathrm{n}}v_{ij}^{n}\right]\boldsymbol{n}_{ij} \tag{4.1.29}$$

[1] https://www.edemsimulation.com
[2] https://www.itascacg.com/software/pfc
[3] https://www.yade-dem.org
[4] https://www.cfdem.com



(a) Visualization of linear spring-dashpot model as parallel spring and dashpot in normal contact of two spherical particles.



(b) Evolution of collision force (4.1.29) for linear spring-dasphot model with (dashed line) and without (solid line) the attractive force during unloading in a binary collision.

Figure 4.3: Visualizations for the linear spring-dashpot contact model.

where a force depending on the relative velocity along the contact normal is introduced. In analogy to the mechanical device which uses viscous friction to provide damping, the force is sometimes called viscous damping force and (4.1.29) is referred to as *linear spring-dashpot model* in the context of DEM (thus the typical visualization shown in Figure 4.3a).

To choose meaningful values for the damping constant d_n it is helpful to consider the binary collision of two particles. As it turns out, the overlap follows a harmonic oscillator equation (see Appendix E) and we can define a damping ratio

$$\zeta_\mathrm{n} = \frac{d_\mathrm{n}}{2\sqrt{m^*_\mathrm{s} k^*_\mathrm{n}}} \tag{4.1.30}$$

which determines the amount of dissipation introduced into the model and whether it prevents a particle rebound. For DEM simulations the relevant range of ζ_n values that still allow for separation of particles after contact is given by

$$\zeta_\mathrm{n} \in [0, 1) \tag{4.1.31}$$

While this dimensionless coefficient already provides an intuition of the amount of damping introduced by the dashpot force, it is not the standard measure of this type in DEM. Instead, it is common to consider the so-called *coefficient of restitution* e_n which is defined as the ratio of velocity before and after the collision, i.e. $e_\mathrm{n} = 1$ for a purely elastic contact and $e_\mathrm{n} = 0$ if the damping prevents separation of particles. In Appendix E it is shown that the damping coefficient and coefficient of restitution for (4.1.29) are related via

$$\zeta_\mathrm{n} = -\frac{\ln(e_\mathrm{n})}{\sqrt{\pi^2 + \ln(e_\mathrm{n})^2}} \tag{4.1.32}$$

Remembering that the damping ratio only depends on effective mass and spring stiffness in the model, we observe that the coefficient of restitution is independent of the collision velocity. In fact, this observation is in contradiction to experimental results, as discussed in [132], but can still be acceptable depending on the range of impact velocities within the considered granular system. This is also backed by the vast pool of literature on DEM which adapts this contact model.

While its easy formulation and analytical results make (4.1.12) a convenient model choice, the plot of force evolution with respect to overlap in Figure 4.3b reveals a problem. The dotted line, representing (4.1.12) and mostly coinciding with the solid line discussed later,

shows a reversal of sign in the contact force during the separation phase (phase of decreasing overlap), which means that attractive forces occur. For non-adhesive particles this effect is typically undesired and eliminated by setting the contact forces to zero as soon as the sign reversal occurs [103, 134, 152]. This results in

$$\boldsymbol{F}^n_{\mathrm{c},ij} = \max\left(0, \left[k^*_{\mathrm{n}}\delta^n_{ij} - d_{\mathrm{n}}v^n_{ij}\right]\right)\boldsymbol{n}_{ij} \tag{4.1.33}$$

and the force-displacement relation is now represented by the solid line in Figure 4.3b. Setting contact forces to zero at an earlier point during the collision means that the above analytical results are not strictly valid any more. In particular, we can expect the contact duration to be overpredicted while the coefficient of restitution is underpredicted. The correct solution to the case without attractive forces has been presented by Schwager and Pöschel [134] by adapting the conditions from which the contact duration is derived. For the readers convenience, the derivation for this modification in our notation can be found in Appendix E. It turns out that with the adapted contact duration, the derivation of an expression for the coefficient of restitution in terms of damping ratio is still possible, while unfortunately the reverse requires a numerical fitting such as the one proposed by Thornton et al. [152].

Damping in Hertz model

After having extended the purely elastic linear spring model to account for dissipative effects, we now want to do the same for the Hertzian model (4.1.25) with varying spring stiffness. First, we want to consider the simplistic extension which is fully analogous to the constant spring model, i.e.

$$\boldsymbol{F}^n_{\mathrm{c},ij} = \left[k^{\mathrm{H}}_{\mathrm{n}}\left(\delta^n_{ij}\right)\delta^n_{ij} - d_{\mathrm{n}}v^n_{ij}\right]\boldsymbol{n}_{ij} \tag{4.1.34}$$

with a constant value of d_{n}. This model was critically discussed by Schäfer et al. [132], who revealed that the combination of Hertzian spring and linear damper produces an increasing coefficient of restitution with increasing collision velocity, which we recall from the linear spring-dashpot model is opposite to the experimentally observed behavior. This is one of the reasons that the above expression only finds rare usage in literature. Instead, as discussed by Anypov and Elliott [5], the two most prevalent models in DEM use a nonlinear damping term leading to either

$$\boldsymbol{F}^n_{\mathrm{c},ij} = \left[k^{\mathrm{H}}_{\mathrm{n}}\left(\delta^n_{ij}\right)\delta^n_{ij} - d_{\mathrm{n}}\sqrt{\delta^n_{ij}}v^n_{ij}\right]\boldsymbol{n}_{ij} \tag{4.1.35}$$

or

$$\boldsymbol{F}^n_{\mathrm{c},ij} = \left[k^{\mathrm{H}}_{\mathrm{n}}\left(\delta^n_{ij}\right)\delta^n_{ij} - d_{\mathrm{n}}\sqrt[4]{\delta^n_{ij}}v^n_{ij}\right]\boldsymbol{n}_{ij} \tag{4.1.36}$$

The former model was proposed by Kuwabare and Kono [85] as well as Brilliantov et al. [12] for collisions of visco-elastic spherical particles and is able to reproduce an increasing coefficient of restitution for increasing collision velocity. The latter model is attributed to Tsuji et al. [157] and results in a constant coefficient of restitution, similar to the linear spring-dashpot model. For a more detailed discussion we chose to restrict ourselves to (4.1.36) which has seen adoption in major implementations of the discrete element method such as the two previously mentioned codes LIGGGHTS and EDEM.

Initially the model was proposed by Tsuji et al. with a constant damping coefficient

$$d_{\mathrm{n}} = \alpha(e_{\mathrm{n}})\sqrt{m^*_{\mathrm{s}}\tilde{k}^{\mathrm{H}}_{\mathrm{n}}} \tag{4.1.37}$$

where we recall that $\tilde{k}^{\mathrm{H}}_{\mathrm{n}}$ denotes the constant part of the Hertzian spring definition. Due to the added complexity of nonlinearities in spring and damper force, the correct analytical relation between d_{n} and e_{n} was not available at the time such that the authors only provided

a graphical representation. Even in more recent articles such as the one by Thornton et al. [152], this relation had only been given as a numerical fit to DEM simulation results. Nevertheless, a solution has been proposed by Anypov and Elliott [5] through the use of a suitable mapping onto the linear spring-dashpot model. The authors found that the correct relation between α and e_n is simply given by

$$\alpha(e_\mathrm{n}) = \sqrt{5}\zeta_\mathrm{n}(e_\mathrm{n}) \tag{4.1.38}$$

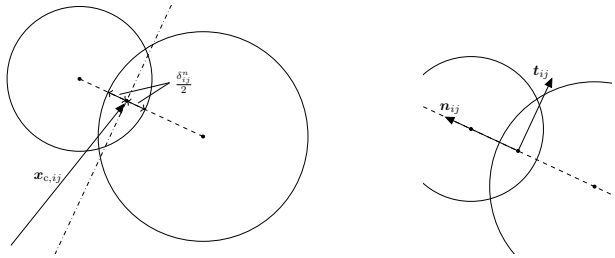
with $\zeta_\mathrm{n}(e_\mathrm{n})$ as defined in (4.1.32). With this result we can, just as easily as in the linear spring-dashpot model, choose a damping constant so that experimentally observed rebound behavior is reproduced. We note that the derivation found in [5] is again based on the model which allows attractive forces. The authors claim that an extension of their result is straight-forward and that in many DEM simulations the attractive forces would not be problematic as long as the correct coefficient of restitution is reproduced. We refrain from following through with the derivation of the non-adhesive analytical relation and refer to the fitted equation by Thornton et al. [152] for this case.

At this point we want to stop the discussion on force expressions along the contact normal direction, since we feel for the purposes of this work the above considerations have conveyed a sufficiently large portion of the general ideas and problems occurring in this field of modeling. The presented contact models will also be sufficient for all of our simulations in the next chapter. For details on further classes of contact models, such as hysteretic spring models, we instead refer the reader to one of the following review articles [38, 81, 152] or the book by O'Sullivan [109] and references therein. The careful reader might have noticed that we specifically limited the end of discussion to normal contacts and in fact we still want to shed some light onto the inclusion of forces acting within the tangent plane in DEM. This will be the subject-matter of the following, in comparison rather short, section. Similar to the previously mentioned treatment of rotations, while work has been done on this topic, the tangential contact models have not yet been sufficiently validated within the context of this work. They were therefore not used in the simulations presented in Chapter 5 such that this subsection should be understood as an outlook and it is up to the reader to skip it.

4.1.3 Tangential contact models

As pointed out in [109, 117], the treatment of tangential forces in DEM as a result of surface asperities or other deviations from perfectly spherical shape reveals a contradiction to the assumptions within the derivation of normal contact models such as the Hertzian spring model. Nevertheless it is common practice to combine the models from last section with fairly similar ones in tangential direction to model experimentally observed frictional effects.

In the original work by Cundall and Strack [29], the authors extended the linear spring model which they employed in normal direction by an analogous model within the tangential contact plane. Clearly, for a fixed point in time there is no immediate geometrical definition of tangential overlap. Instead, the authors proposed to take the time integral of a tangential velocity to obtain a measure similar to the normal overlap. To properly introduce such a model there are two points which warrant a closer look. Firstly, since we do not restrict ourselves to two dimensional problems, the definition of tangential unit vector is not immediately clear and might be any normalized vector in the contact plane defined by the normal direction. Secondly, we settled for an application of the force directly to the particle centroids in the above introduction to normal contact models. As one might imagine, when we want to consider tangential forces, which primarily aim to model friction between the spheres, the corresponding forces should be applied to the (deformed) particle surfaces where



(a) Sketch of contact point location as defined in (4.1.39) and corresponding tangential plane (dash-dotted line).

(b) Sketch of normal and tangential unit vectors.

Figure 4.4: Visualization of contact point and unit vectors in the collision of two spherical particles.

the frictional stresses apply. This is typically done by defining a single point within the contact volumes that is common to both particles and where the forces are applied. This point, which is often called *contact point* of the collisions, does in fact not have a unique definition in literature, as is pointed out by Lu et al. [96]. The authors even note that it might be necessary to distribute among a number of contact points to reproduce physically reasonable behavior for non-spherical particle contacts. For spherical particles it is however customary [41, 97, 109, 139] to define the contact point as the middle point of the overlapping volume along the normal direction, i.e.

$$\boldsymbol{x}_{c,ij} = \boldsymbol{x}_{s,j} + \left(R_{s,j} - \frac{\delta_{ij}^n}{2}\right)\boldsymbol{n}_{ij} \tag{4.1.39}$$

$$\boldsymbol{x}_{c,ij} = \boldsymbol{x}_{s,i} - \left(R_{s,i} - \frac{\delta_{ij}^n}{2}\right)\boldsymbol{n}_{ij} \tag{4.1.40}$$

This definition is also visualized in Figure 4.4a. From the sketch it is already evident that this definition does, for different particle radii, not coincide with the centroid of the line or circular region which are bounded by the intersection of particle surfaces. Before moving forward to a relative velocity definition, we want to simplify notations by introducing the so-called *reduced radii*

$$R'_{s,i} = R_{s,i} - \frac{\delta_{ij}^n}{2} \tag{4.1.41}$$

which correspond to the radii of spheres centered in $\boldsymbol{x}_{s,i}$, $\boldsymbol{x}_{s,j}$ that touch in $\boldsymbol{x}_{c,ij}$. With this definition we write the velocities due to rigid body motion of the two material points, which coincide in world frame coordinates (see Appendix D) with the contact point, as

$$\boldsymbol{v}_{c,j} = \boldsymbol{v}_{s,j} + \boldsymbol{\omega}_{s,j} \times (\boldsymbol{x}_{c,ij} - \boldsymbol{x}_{s,j}) = \boldsymbol{v}_{s,j} + \boldsymbol{\omega}_{s,j} \times \left(R'_{s,j}\boldsymbol{n}_{ij}\right) \tag{4.1.42}$$

$$\boldsymbol{v}_{c,i} = \boldsymbol{v}_{s,i} + \boldsymbol{\omega}_{s,i} \times (\boldsymbol{x}_{c,ij} - \boldsymbol{x}_{s,i}) = \boldsymbol{v}_{s,i} - \boldsymbol{\omega}_{s,i} \times \left(R'_{s,i}\boldsymbol{n}_{ij}\right) \tag{4.1.43}$$

where $\boldsymbol{\omega}_{s,i}$, $\boldsymbol{\omega}_{s,j}$ denote the angular velocities of the particles (in world frame). The relative velocity at the contact point are then defined as

$$\boldsymbol{v}_{c,ij} = \boldsymbol{v}_{c,i} - \boldsymbol{v}_{c,j} = \boldsymbol{v}_{s,i} - \boldsymbol{v}_{s,j} - \left(R'_{s,i}\boldsymbol{\omega}_{s,i} + R'_{s,j}\boldsymbol{\omega}_{s,j}\right) \times \boldsymbol{n}_{ij} \tag{4.1.44}$$

We notice that upon taking the dot product of this velocity with $\boldsymbol{n}_{ij}$ the rotational term vanishes and the normal component of this velocity reduces to $v_{s,ij}^n$, as in (4.1.10). From the remaining components of velocity one can define a tangential velocity

$$\boldsymbol{v}_{s,ij}^t = \boldsymbol{v}_{s,ij} - v_{s,ij}^n \boldsymbol{n}_{ij} \tag{4.1.45}$$

The simplest model for frictional forces that has seen application in molecular dynamics and DEM is purely based on this relative velocity. It is sometimes referred to as *Haff-Werner friction* [117] and is given by

$$\boldsymbol{F}_{\mathrm{c},ij}^{t} = -\min\left(\mu\left\|\boldsymbol{F}_{\mathrm{c},ij}^{n}\right\|, d_{\mathrm{t}}\left\|\boldsymbol{v}_{\mathrm{s},ij}^{t}\right\|\right)\boldsymbol{t}_{ij} \tag{4.1.46}$$

with tangential unit vector

$$\boldsymbol{t}_{ij} = \frac{\boldsymbol{v}_{\mathrm{s},ij}^{t}}{\left\|\boldsymbol{v}_{\mathrm{s},ij}^{t}\right\|} \tag{4.1.47}$$

and sliding friction coefficient μ. Clearly, such a purely dissipative force is not able to support static configurations such as heaps (see also [103]), but according to Pöschel and Schwager [117] the method has still seen successful application in molecular dynamics dominated by contacts of finite duration.

To obtain a model which supports the case of static friction, we can instead introduce the analogue to normal overlap

$$\boldsymbol{\xi}_{ij}^{t} = \int_{t_{ij}^{0}}^{t} \boldsymbol{v}_{\mathrm{s},ij}^{t}\,\mathrm{d}t \tag{4.1.48}$$

which is often referred to as *tangential displacement* or *directional spring deformation*. In this definition t_{ij}^{0} denotes the time at which the particles i and j first came into contact. A tangential spring model, similar to that proposed by Cundall and Strack, can then be written as

$$\boldsymbol{F}_{\mathrm{c},ij}^{t} = -k_{\mathrm{t}}^{*}\boldsymbol{\xi}_{ij}^{t} \tag{4.1.49}$$

While in normal direction the spring force aims to restore the state of zero overlap, i.e. non-penetrating particles, the tangential force is supposed to model the static friction between particles, e.g. the interlocking of asperities on particle surfaces. Analogous to the purely dissipative force presented above, the spring force is typically limited by Coulomb friction, i.e.

$$\boldsymbol{F}_{\mathrm{c},ij}^{t} = \min\left(\mu\left\|\boldsymbol{F}_{\mathrm{c},ij}^{n}\right\|, \left\|\boldsymbol{F}_{\mathrm{c},ij}^{t}\right\|\right)\boldsymbol{t}_{ij} \tag{4.1.50}$$

for a friction coefficient μ (for simplicity we assumed identical static and sliding friction coefficients) and tangential unit vector

$$\boldsymbol{t}_{ij} = \frac{\boldsymbol{F}_{\mathrm{c},ij}^{t}}{\left\|\boldsymbol{F}_{\mathrm{c},ij}^{t}\right\|} \tag{4.1.51}$$

There are some more intricacies to this model, such as projections to keep the directional spring within tangential planes, which we do not discuss here. For an easy introduction to this subject-matter we refer the reader to the article by Luding [97].

Similar to the preceding section we could now elaborate on a wide range of extensions to this simple spring model, some of which also increment the tangential force instead of the quantity in (4.1.48). Since the goal of this short section was only to convey the general idea of tangential contact models, we want to conclude this section with references to more complete reviews that the interested reader may consult. Apart from the previously cited book by O'Sullivan [109] we specifically refer to [38, 41, 80, 152]. We further want to note that one can define other relative velocities to specifically treat rolling friction or friction due to torsion [1, 69, 74, 97]. While the former one is used frequently in DEM, the latter one is rarely considered [109].

4.1.4 Calculation of coupling forces

Aside from forces due to particle contacts, we also want to calculate the coupling forces discussed in Section 3.4. Here we explain how this is realized in our generalized finite difference framework. Clearly, in the particle momentum equation (4.1.2) these terms are subsumed within the body force $\boldsymbol{F}_{\mathrm{b},i}$. First, let us consider the pressure gradient and viscous stress force

$$\boldsymbol{F}_{\mathrm{p},i} = -\left(\boldsymbol{\nabla} p\right)|_{\boldsymbol{x}_{\mathrm{s},i}} V_{\mathrm{s},i} \tag{4.1.52}$$

$$\boldsymbol{F}_{\tau,i} = \left(\boldsymbol{\nabla} \cdot \boldsymbol{\tau}_{\mathrm{f}}\right)|_{\boldsymbol{x}_{\mathrm{s},i}} V_{\mathrm{s},i} \tag{4.1.53}$$

Note that we omitted the averaging notation since, at this point, it should be clear that we solve for continuous phase quantities in our numerical scheme that correspond to averages in the theoretical framework. In Chapter 2 we discussed how to construct differential operators at a point location from values of neighboring points through a least squares approach. Clearly, as can be done when a new point is added, we can determine these stencils also at any other location within the domain discretized by the point cloud. In particular, for a particle with position $\boldsymbol{x}_{\mathrm{s},i}$, we can determine the (indices of) fluid phase neighbor points

$$\mathcal{N}_{\mathrm{f}}(\boldsymbol{x}_{\mathrm{s},i}) = \{j \mid j \in \{1, \dots, N_{\mathrm{f}}\},\ \|\boldsymbol{x}_{\mathrm{f},j} - \boldsymbol{x}_{\mathrm{s},i}\| \leq h_{\mathrm{f},i}\} \tag{4.1.54}$$

where the smoothing length $h_{\mathrm{f},i}$ has to be chosen based on the local smoothing lengths of surrounding fluid points. However, this neighbor list has to be constructed for every particle, which quickly becomes prohibitively costly in terms of computation time. Instead, for every particle, the closest fluid point is determined and its already available neighbor list is used to compute stencils at the solid particle position. Throughout this work we have used monomials up to an order $\mathrm{P} = 2$ to determine the first order stencils $\boldsymbol{c}_i^x$, $\boldsymbol{c}_i^y$, $\boldsymbol{c}_i^z$. Then the gradient is obtained by applying these stencils component-wise, i.e.

$$\left(\boldsymbol{\nabla} p\right)|_{\boldsymbol{x}_{\mathrm{s},i}} \approx \begin{bmatrix} \sum\limits_{j \in \mathcal{N}_{\mathrm{f},k}} c_{ij}^x p_j \\ \sum\limits_{j \in \mathcal{N}_{\mathrm{f},k}} c_{ij}^y p_j \\ \sum\limits_{j \in \mathcal{N}_{\mathrm{f},k}} c_{ij}^z p_j \end{bmatrix}, \quad k = \underset{l \in \{1,\dots,N_{\mathrm{f}}\}}{\arg\min} \|\boldsymbol{x}_{\mathrm{f},l} - \boldsymbol{x}_{\mathrm{s},i}\| \tag{4.1.55}$$

We recall from (3.2.96), that the viscous stress is given by

$$\boldsymbol{\tau}_{\mathrm{f}} = \mu\Big(\boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}} + \left(\boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}}\right)^T - \frac{2}{3}(\boldsymbol{\nabla} \cdot \boldsymbol{u}_{\mathrm{f}})\mathbb{1}\Big) \tag{4.1.56}$$

with a constant dynamic viscosity μ. Using the identities in Appendix A, it is easy to verify that this leads to

$$\boldsymbol{F}_{\tau} = \mu\Big(\boldsymbol{\nabla} \cdot \left(\boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}}\right) + \frac{1}{3}\boldsymbol{\nabla} \cdot \left(\boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}}\right)^T\Big) V_{\mathrm{s}} \tag{4.1.57}$$

The second order derivatives within this expression are calculated in two steps. First, each component of the velocity gradient is calculated in the same way as shown above for the pressure gradient. These values are stored and in a second step the divergence terms are calculated by applying the stencils again to the stored gradient and its transposed.

After having dealt with the contributions due to the resolved part of fluid stress, we now want to turn our attention towards the only expression that is used for the unresolved stress within this work, namely drag force. In Section 3.4 we saw that the calculation of this force only requires the knowledge of fluid velocity and porosity at the particle position. Assuming that both quantities are available in the fluid point cloud, these values at $\boldsymbol{x}_{\mathrm{s},i}$ are easily

obtained by applying the function approximation stencil $\boldsymbol{c}_i^0$, which is determined analogously to the above first order derivative stencils. With the order of calculations presented at the very beginning of this chapter (see Figure 4.1) it is clear, that all the values attached to fluid points correspond to either initial conditions or the previous time step. Thus, the way that these values are determined will be detailed in the upcoming Section 4.2 and Section 4.4.

4.1.5 Time integration

Now that we have introduced all the forces which are expected to form the right-hand side of (4.1.2), we briefly want to discuss the numerical integration of (4.1.1), (4.1.2). To simplify notations in the following, we define the particle acceleration

$$\boldsymbol{a}_{\mathrm{s},i}(t) = \frac{1}{m_{\mathrm{s},i}} \boldsymbol{F}_{\mathrm{b},i}(t) + \frac{1}{m_{\mathrm{s},i}} \sum_{j=1}^{N_\mathrm{s}} \boldsymbol{F}_{\mathrm{c},ij}(t) \tag{4.1.58}$$

so that (4.1.2) becomes

$$\frac{\mathrm{d}\boldsymbol{v}_{\mathrm{s},i}}{\mathrm{d}t} = \boldsymbol{a}_{\mathrm{s},i}(t) \tag{4.1.59}$$

We point out that the body force $\boldsymbol{F}_{\mathrm{b},i}$ typically depends only on the position and velocity of particle i, i.e.

$$\boldsymbol{F}_{\mathrm{b},i}(t) = \boldsymbol{F}_{\mathrm{b},i}(t, \boldsymbol{x}_{\mathrm{s},i}(t), \boldsymbol{v}_{\mathrm{s},i}(t)) \tag{4.1.60}$$

while the sum of collision forces formally depends on the position and velocity of all particles, i.e.

$$\boldsymbol{F}_{\mathrm{c},i}(t) = \boldsymbol{F}_{\mathrm{c},i}(t, \boldsymbol{X}_\mathrm{s}(t), \boldsymbol{V}_\mathrm{s}(t)) \tag{4.1.61}$$

with

$$\boldsymbol{X}_\mathrm{s}(t) = \{\boldsymbol{x}_{\mathrm{s},i}(t) \mid i = 1, \ldots, N_\mathrm{s}\}, \qquad \boldsymbol{V}_\mathrm{s}(t) = \{\boldsymbol{v}_{\mathrm{s},i}(t) \mid i = 1, \ldots, N_\mathrm{s}\} \tag{4.1.62}$$

We note however, that for particles of spherical shape and with the small overlaps required in DEM, the number of neighbors for each particle is rather small. In fact for equally sized spheres an upper bound is formed by the kissing number of 6 in two dimensions and 12 in three dimensions. Nevertheless, we use the general form

$$\boldsymbol{a}_{\mathrm{s},i}(t) = \boldsymbol{a}_{\mathrm{s},i}(t, \boldsymbol{X}_\mathrm{s}(t), \boldsymbol{V}_\mathrm{s}(t)) \tag{4.1.63}$$

in the following. Reviews of different time integration schemes can be found in the articles by Dziugys and Peters [41], Fraige and Langston [51] as well as Kruggel-Emden et al. [82]. Since for different applications of the discrete element method the restrictions on memory requirement, number of force evaluations and computational cost of a time integration scheme might differ, the authors in [82] recommend the usage of rather simple schemes such as second or third order Taylor expansion schemes for universal application. More complex methods can certainly still be beneficial in cases where the above restrictions are comparatively loose, but since both, in [82] and [51], good results are obtained from the simple second order Taylor expansion and half-step leapfrog Verlet scheme, we chose a set of rather simple integrators for our investigations with the possibility of later extensions in mind. In particular, this does not include any implicit integration schemes, which are rather rare in the DEM context. An argument for the usage of less complex integration schemes can also be made from the fact that Kruggel-Emden et al. [79], in their follow-up article to [82], found that the choice of integrator had little impact on macroscopic observables such as the change of total mass in a hopper or the velocity profile of a particle assembly. Let us now specify the schemes which have been considered within the context of this work. We note that, in

order to simplify later reference to any of the following schemes, we introduce them together with shorthands which can be found in brackets behind their respective name.

Explicit / forward Euler scheme (EE)
The first scheme which comes to mind, obviously is the classical first order *forward* or *explicit Euler* scheme given by

$$\boldsymbol{x}_\mathrm{s}(t+\Delta t) = \boldsymbol{x}_\mathrm{s}(t) + \boldsymbol{v}_\mathrm{s}(t)\Delta t \tag{4.1.64}$$
$$\boldsymbol{v}_\mathrm{s}(t+\Delta t) = \boldsymbol{v}_\mathrm{s}(t) + \boldsymbol{a}_\mathrm{s}(t, \boldsymbol{X}_\mathrm{s}(t), \boldsymbol{V}_\mathrm{s}(t))\Delta t \tag{4.1.65}$$

The explicit Euler scheme has for example been used in the previously mentioned paper by Tsuji et al. [157], but it is well known for providing poor stability.

Symplectic / semi-implicit Euler scheme (SE)
In [31, 158] the authors consider the *symplectic* or *semi-implicit Euler scheme* of the form

$$\boldsymbol{v}_\mathrm{s}(t+\Delta t) = \boldsymbol{v}_\mathrm{s}(t) + \boldsymbol{a}_\mathrm{s}(t, \boldsymbol{X}_\mathrm{s}(t), \boldsymbol{V}_\mathrm{s}(t))\Delta t \tag{4.1.66}$$
$$\boldsymbol{x}_\mathrm{s}(t+\Delta t) = \boldsymbol{x}_\mathrm{s}(t) + \boldsymbol{v}_\mathrm{s}(t+\Delta t)\Delta t \tag{4.1.67}$$

which is the result of forward Euler applied to velocity and backward Euler to the position. This scheme is still first order but has the advantage of being a symplectic integrator. The authors observed this integration method to be the most accurate and robust first order scheme for both, linear spring-dashpot and Hertzian models. They further recommended this scheme for usage in the context of CFD-DEM.

Second order Taylor scheme (TY2)
A second order Taylor expansion scheme is described by Dziugys and Peters [41] and Kruggel-Emden et al. [82]. It is given by

$$\boldsymbol{x}_\mathrm{s}(t+\Delta t) = \boldsymbol{x}_\mathrm{s}(t) + \boldsymbol{v}_\mathrm{s}(t)\Delta t + \frac{1}{2}\boldsymbol{a}_\mathrm{s}(t, \boldsymbol{X}_\mathrm{s}(t), \boldsymbol{V}_\mathrm{s}(t))\Delta t^2 \tag{4.1.68}$$
$$\boldsymbol{v}_\mathrm{s}(t+\Delta t) = \boldsymbol{v}_\mathrm{s}(t) + \boldsymbol{a}_\mathrm{s}(t, \boldsymbol{X}_\mathrm{s}(t), \boldsymbol{V}_\mathrm{s}(t))\Delta t \tag{4.1.69}$$

In [51] instead

$$\boldsymbol{v}_\mathrm{s}(t+\Delta t) = \boldsymbol{v}_\mathrm{s}(t) + \boldsymbol{a}_\mathrm{s}(t, \boldsymbol{X}_\mathrm{s}(t), \boldsymbol{V}_\mathrm{s}(t))\Delta t \tag{4.1.70}$$
$$\boldsymbol{x}_\mathrm{s}(t+\Delta t) = \boldsymbol{x}_\mathrm{s}(t) + \frac{\boldsymbol{v}_\mathrm{s}(t+\Delta t) + \boldsymbol{v}_\mathrm{s}(t)}{2}\Delta t \tag{4.1.71}$$

is denoted as modified Euler scheme. Plugging the $\boldsymbol{v}_\mathrm{s}(t+\Delta t)$ from (4.1.70) into (4.1.71) reveals the equality to the second order Taylor scheme. This scheme has a truncation error of $\mathcal{O}(\Delta t^3)$ in the position but $\mathcal{O}(\Delta t^2)$ in velocity, so it is still first order in velocity.

Central difference / velocity Verlet scheme (CD)
One of the standard integration methods in DEM is given by the second order *velocity Verlet* scheme, also known as *central difference* scheme, which is of the form [41, 51, 70, 82, 88, 126]

$$\boldsymbol{v}_\mathrm{s}(t+\frac{\Delta t}{2}) = \boldsymbol{v}_\mathrm{s}(t-\frac{\Delta t}{2}) + \boldsymbol{a}_\mathrm{s}(t)\Delta t$$
$$\boldsymbol{x}_\mathrm{s}(t+\Delta t) = \boldsymbol{x}_\mathrm{s}(t) + \boldsymbol{v}_\mathrm{s}(t+\frac{\Delta t}{2})\Delta t$$

We note that this scheme is also sometimes referred to as *(half-step) leap-frog Verlet* [41, 51]. This naming convention is not adopted in the following since it appears to be ambiguous, as can be seen from [82] where it refers to the slightly different position Verlet scheme. Hanley et al. [70] note that this scheme has also been widely adopted in open-source DEM codes like the previously mentioned LIGGGHTS and Yade.

We observe that the position and velocity are calculated at shifted points in time. This introduces an additional complication if we want to consider velocity-dependent forces. In many articles [41, 88, 126] such forces are not considered or the necessary modification is not explicitly given. Fraige and Langston [51] on the other hand propose the additional step

$$\boldsymbol{v}_\mathrm{s}(t+\Delta t) = \boldsymbol{v}_\mathrm{s}(t+\frac{\Delta t}{2}) + \frac{1}{2}\boldsymbol{a}_\mathrm{s}(t)\Delta t \tag{4.1.72}$$

which is executed after the above calculations, meaning that in their case

$$\boldsymbol{a}_\mathrm{s}(t) = \boldsymbol{a}_\mathrm{s}(t, \boldsymbol{X}_\mathrm{s}(t), \boldsymbol{V}_\mathrm{s}(t)) \tag{4.1.73}$$

They found this scheme to be more accurate and stable than what they call the modified Euler scheme, which is the above second order Taylor scheme in our naming convention. In [29, 70, 82] the authors instead use positions and velocities which are half a time step apart for the evaluation of forces, i.e.

$$\boldsymbol{a}_\mathrm{s}(t) = \boldsymbol{a}_\mathrm{s}(t, \boldsymbol{X}_\mathrm{s}(t), \boldsymbol{V}_\mathrm{s}(t-\frac{\Delta t}{2})) \tag{4.1.74}$$

Since the initial conditions of position and velocity are typically given at the same point in time, $t = 0$ for simplicity, we further need to prescribe a value for $\boldsymbol{v}_\mathrm{s}(-\frac{\Delta t}{2})$ in the first time step. Here we employ

$$\boldsymbol{v}_\mathrm{s}(-\frac{\Delta t}{2}) = \boldsymbol{v}_\mathrm{s}(0) - \frac{1}{2}\boldsymbol{a}_\mathrm{s}(0, \boldsymbol{X}_\mathrm{s}(0), \boldsymbol{V}_\mathrm{s}(0))\Delta t \tag{4.1.75}$$

Now that we have introduced a set of integration schemes, we want to do two things. First, we discuss time step restrictions which are typically imposed and introduce the necessary notations for our second step. This will be a short analysis of the proposed schemes for one of the standard test cases in DEM, the collision of a single particle with a wall. We consider this test case for both, the linear spring-dashpot model and the damped Hertzian model of type (4.1.36). This serves as an elementary unit test of our DEM scheme and we use this to decide on the numerical integration scheme which will be used in the following. We note that, at this point, only time step restrictions due to the DEM contact model are discussed. The influence of drag forces is postponed to a later point within this chapter, namely Section 4.7.

Time step restrictions

In discrete element methods it is customary to postulate an upper bound for the time step of the form [70, 99]

$$\Delta t_\mathrm{s} \leq C t_\mathrm{c} \tag{4.1.76}$$

where C is a constant which is typically chosen smaller than one and

$$t_\mathrm{c} = \sqrt{\frac{m_\mathrm{s}^*}{k_\mathrm{n}^*}} \tag{4.1.77}$$

is the critical time in the harmonic oscillator analogue to the linear spring-dashpot contact model described in Appendix E. A discussion on the choice of C can be found in the article

by Malone and Xu [99]. There the authors point out that a wide range of values have been proposed in DEM literature. Using the central difference scheme and purely elastic linear spring model they further numerically determined a value of $C = 0.2$, which they found is necessary to obtain less than 5% error in potential energy after collision of a single particle with a wall under the effect of gravity. For two colliding particles with attractive forces allowed during separation we derived in Appendix E that the contact duration is given by

$$t_\mathrm{d} = \frac{\pi}{\sqrt{1-\zeta_\mathrm{n}^2}} t_\mathrm{c} \tag{4.1.78}$$

For the case without attractive forces one instead obtains (see Appendix E)

$$t_\mathrm{d} = \frac{t_\mathrm{c}}{\sqrt{1-\zeta_\mathrm{n}^2}} \begin{cases} \pi - \operatorname{atan}\left(\dfrac{2\zeta_\mathrm{n}\sqrt{1-\zeta_\mathrm{n}^2}}{2\zeta_\mathrm{n}^2-1}\right) & \zeta_\mathrm{n} < \dfrac{1}{\sqrt{2}} \\ \operatorname{atan}\left(\dfrac{2\zeta_\mathrm{n}\sqrt{1-\zeta_\mathrm{n}^2}}{2\zeta_\mathrm{n}^2-1}\right) & \zeta_\mathrm{n} > \dfrac{1}{\sqrt{2}} \end{cases} \tag{4.1.79}$$

Evidently, in both cases the damping increases the contact duration and without damping both expressions reduce to the same value of $t_\mathrm{d} = \pi t_\mathrm{c}$. This means that the suggested value of $C = 0.2$ leads to at least 15 integration steps within the collision duration. Another investigation on this subject-matter has been published by O'Sullivan and Bray [110] who used methods from finite element analysis and considered the central difference scheme and a linear elastic contact model. For a selection of sphere arrangements in 3D they found the conservative bound

$$\Delta t_\mathrm{s} \leq 0.17\sqrt{\frac{m_\mathrm{s,min}}{k_\mathrm{n,max}}} \tag{4.1.80}$$

where the minimum and maximum are taken over the complete system of particles. Otsubo et al. [112] later extended the considerations from [110] to more realistic particle configurations and proposed the bound

$$\Delta t_\mathrm{s} \leq \sqrt{\frac{m_\mathrm{s,min}}{C_\mathrm{N,max} k_\mathrm{n,max}}} \tag{4.1.81}$$

instead. Here we denote by $C_\mathrm{N,max}$ the maximum coordination number, i.e. number of neighbors that a particle interacts with, among all particles. Obviously, $C_\mathrm{N,max}$ is bounded by the above mentioned kissing number for spherical particles of equal size in 3D while it might be significantly larger for polydisperse assemblies. As a conservative bound across all coordination numbers in their particle configurations they found

$$\Delta t_\mathrm{s} \leq 0.1\sqrt{\frac{m_\mathrm{s,min}}{k_\mathrm{n,max}}} \tag{4.1.82}$$

in three dimensions. They note that in many cases such a choice might be over conservative and waste computational resources. They further discuss other means of obtaining estimates for the critical time step which forms the upper bound to the integration time step.

Finally, for the contact of two spherical particles with a purely elastic Hertzian spring model one can analytically derive the duration of the collision to be [86, 116]

$$t_\mathrm{d} = 2.94\left(\frac{15}{16}\frac{m_\mathrm{s}^*}{\Upsilon^*\sqrt{R_\mathrm{s}^*}}\right)^{\frac{2}{5}}\left(\frac{1}{v_{\mathrm{s},ij}^{n,0}}\right)^{\frac{1}{5}} \tag{4.1.83}$$

We notice that in contrast to the linear spring-dashpot model the contact duration explicitly depends on the magnitude of relative velocity at the beginning of contact, again denoted

by $v_{\mathrm{s},ij}^{n,0}$. Consequently, in order to choose the time step as a specified fraction of contact duration one needs to estimate the occurring collision velocities within the system. We recall that the same had to be done to calculate suitable spring stiffnesses from a maximum allowed overlap in both, the linear and Hertzian model. Thus, even though the equation (4.1.77) might conceal this fact, there is an implicit dependency of the contact duration on the maximum possible value of $v_{\mathrm{s},ij}^{n,0}$ in the system. To be precise, choosing k_{n}^{*} equal to the right-hand side of (4.1.13) leads to

$$t_{\mathrm{d}} = \pi \frac{\delta_{ij}^{\mathrm{n,max}}}{v_{\mathrm{s},ij}^{n,0}} \tag{4.1.84}$$

in a purely elastic collision. Similarly, if we combine (4.1.28) with (4.1.83), we obtain

$$t_{\mathrm{d}} = 2.94 \frac{\delta_{ij}^{\mathrm{n,max}}}{v_{\mathrm{s},ij}^{n,0}} \tag{4.1.85}$$

From this we can conclude that the choice of contact model between linear and Hertzian spring has no significant influence on the time step restriction, as long as they are both chosen based on the same maximum overlap and estimate of collision velocity. We further derived from this the choice

$$\Delta t_{\mathrm{s}} \leq \frac{t_{\mathrm{d}}}{N_{\mathrm{t,d}}}, \quad N_{\mathrm{t,d}} \geq 15 \tag{4.1.86}$$

which we use within this work for the Hertzian model and which we previously obtained from $\Delta t_{\mathrm{s}} \leq C t_{\mathrm{c}}$ and $C = 0.2$ in the linear model. If instead we compare (4.1.85) to the model based on material parameters, we observe a difference in scaling with respect to velocity. As a consequence the contact duration in the Hertzian model with material parameters is significantly smaller at low impact velocities. This is due to the fact, that this model represents very stiff particles. In less dynamic systems it might thus be advantageous to replace the material model by parameters which allow for more overlap in favor of larger time steps.

As for the damped Hertzian model of the form (4.1.36), a numerical investigation of contact duration in this case can be found for example in the article by Antypov and Elliott [5]. From the data they provide it is clear that, at least for $e_{\mathrm{n}} \geq 0.2$, the influence is rather small compared to the large pre-factor which typically applied when setting an upper bound for the time step based on the contact duration.

Naturally, from this discussion, the question arises whether there are theoretical restrictions accompanying these numerically determined time step choices. By employing the harmonic oscillator analogue, one can easily prove by standard stability analysis, that the explicit Euler scheme applied to the spring-damper system is stable as long as

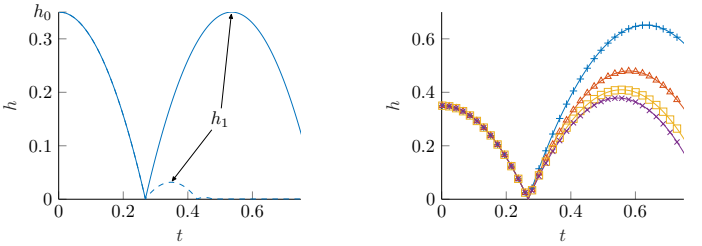
$$\Delta t_{\mathrm{s}} \leq 2\zeta_{\mathrm{n}} t_{\mathrm{c}} \tag{4.1.87}$$

which shows that no stability can be achieved in continued damped contacts, independent of the time step. On the other hand, for the semi-implicit Euler scheme one can prove that stability in the harmonic oscillator integration is ensured as long as

$$\Delta t_{\mathrm{s}} \leq 2\Big(\sqrt{\zeta_{\mathrm{n}}^2 + 1} - \zeta_{\mathrm{n}}\Big) t_{\mathrm{c}} \tag{4.1.88}$$

which reduces to $\Delta t_{\mathrm{s}} \leq 2 t_{\mathrm{c}}$ without damping. We observe that the time step choices detailed above are often stricter than those imposed by the theoretical stability bounds in the harmonic oscillator analogue. On the one hand, this is due to the fact, that the contact model deviates from the oscillator analogue by its discontinuous nature, since interaction only takes places while the particles virtually overlap. On the other hand, stability alone is not enough to obtain physically meaningful results, instead a certain degree of approximation quality is desired when setting up a collision model from experimentally determined parameters.

Benchmark of integration schemes



(a) Particle height h over time in purely elastic (solid line) and strongly damped ($e_\mathrm{n} = 0.3$, dashed line) case. Obtained from central difference scheme with $N_\mathrm{t,c} = 100$.

(b) Particle height h over time from explicit Euler time integration and purely elastic collision. Lines correspond to $N_\mathrm{t,c} = 5$ (+), $N_\mathrm{t,c} = 10$ (triangle), $N_\mathrm{t,c} = 20$ (square), $N_\mathrm{t,c} = 40$ (x).

Figure 4.5: Particle trajectories for the wall collision with linear spring-dashpot contact model. $N_\mathrm{t,c}$ denotes the number of time steps within the critical time t_c, i.e. $t_\mathrm{c} = N_\mathrm{t,c}\Delta t_\mathrm{s}$.

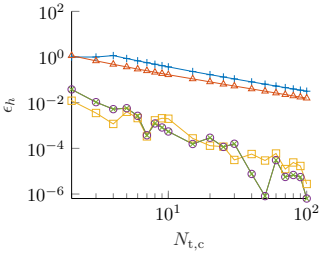
In the following we want to provide a preliminary investigation on the performance of these different time integration scheme in the context of a typical scenario in DEM, the collision of a single particle with a wall. To this end, we chose to use the same setup as in the article by Malone and Xu [99] and extend their investigations. The setup consists of a spherical particle of radius $R_\mathrm{s} = 0.5\,\mathrm{mm}$ and material density $\rho_\mathrm{s} = 2750\,\frac{\mathrm{kg}}{\mathrm{m}^3}$ which is initially placed at a height of $h_0 = 0.35\,\mathrm{m}$ from the wall and moves under the influence of gravity with $g = 9.81\,\frac{\mathrm{m}}{\mathrm{s}^2}$. While the authors in [99] only considered the perfectly elastic liner spring-dashpot model, we also consider a strongly damped version of this model (4.1.29) and repeat the same for the Hertzian model (4.1.36). In the Hertzian model we set the Poisson ratio ν to zero and chose the Young modulus Y so that the theoretically obtained maximum overlap in the Hertzian contact is identical to the linear model. As we saw in (4.1.84) and (4.1.85) this leads to almost the same contact duration in both models. Due to this fact we also test the same time step sizes in both contact models. To be precise, we choose them as fractions

$$\Delta t_\mathrm{s} = \frac{t_\mathrm{c}}{N_\mathrm{t,c}}, \quad N_\mathrm{t,c} \in \mathbb{N}, \quad N_\mathrm{t,c} > 1 \tag{4.1.89}$$

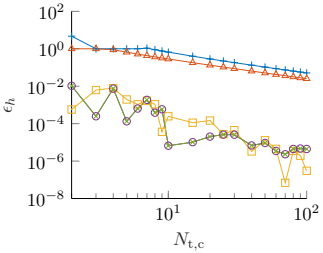
of the critical time t_c in the linear spring-dashpot model. To evaluate the results, we compare the maximum height that the particle reaches after a single collision h_1^sim (see Figure 4.5a) in the simulations against the theoretical value h_1^theo obtained from the relation between pre-collisional and post-collisional velocity $v_1 = e_\mathrm{n} v_0$ and integration of the equations of motion.

When presenting the results we refer to the schemes via their short-hands introduced above. For the central difference scheme in particular, we denote by CD1 the scheme based on (4.1.72), (4.1.73) while CD2 denotes the one which uses the force evaluation at half-step velocities (4.1.74).

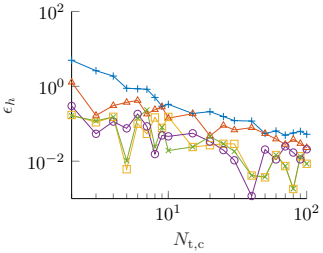
The results for the two purely elastic models are shown in figures 4.6a, 4.6b. Both, the explicit Euler (EE) scheme and the second order Taylor (TY2) scheme, lead to far worse results and can not match the accuracy of the other schemes in this case. This lack of accuracy manifests itself in an increase of kinetic energy across the collision and a subsequently larger maximum height. This is visualized for the explicit Euler scheme



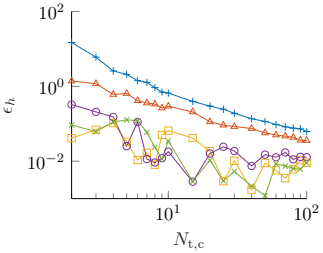
(a) Purely elastic linear spring model.



(b) Purely elastic Hertzian model.



(c) Linear spring-dashpot model, $e_{\mathrm{n}} = 0.3$.



(d) Damped Hertzian model, $e_{\mathrm{n}} = 0.3$.

Figure 4.6: Comparison of DEM time integration schemes for single particle wall impact. The plots show the relative error $\epsilon_h = \left|h_1^{\mathrm{sim}} - h_1^{\mathrm{theo}}\right| / h_1^{\mathrm{theo}}$ over the number of time steps within the critical time t_{c}, i.e. $t_{\mathrm{c}} = N_{\mathrm{t,c}} \Delta t_{\mathrm{s}}$. The lines correspond to the presented schemes with short-hands EE (+), TY2 (triangle), SE (square), CD1 (circle), CD2 (x).

in Figure 4.5b. Among the remaining methods, the two versions of the central difference scheme lead to identical results while the semi-implicit Euler (SE) scheme also shows an overall similar trend. All of these three schemes manage to reproduce the analytical result with less than 1% error for a subdivision of the critical time into at least three time steps, i.e. $N_{\mathrm{t,c}} \geq 3$.

The results of the strongly damped cases are shown in figures 4.6c, 4.6d. While the explicit Euler and Taylor scheme still show larger errors, especially in the Hertzian model, the schemes are not separated as clearly any more and the overall errors are significantly larger, even for small time steps sizes. These rather inconclusive results require a more in-depth investigation which is left open for future work at this point.

Overall we feel confident in concluding that replacing the explicit Euler scheme by the Taylor scheme does not provide any benefit in terms of accuracy and is therefore not worthwhile. Instead, either one of the remaining schemes is expected to improve the stability within a coupled system containing large amounts of particles significantly. This is due to the reduced artificial generation of kinetic energy. From the presented results no obvious conclusion can be drawn on whether any of the central difference schemes or the semi-implicit Euler should be used in future studies. However, since the scheme with force evaluations based on half-step velocities (CD2) did not produce more accurate results within our specific tests, but also introduces the conceptual difficulty of desynchronized time stepping in fluid and solid phase velocities, we drop it in favor of the SE and CD1 schemes until more conclusive results are available.

4.2 Coarse graining

With all aspects of the DEM scheme being clarified, we now proceed to explain the second component within the rough outline of our coupling algorithm, which we showed in Figure 4.1. The goal of this step is the calculation of averaged quantities, such as the volume fraction ε_s and coupling force density $\boldsymbol{f}_\mathrm{fs}$, on the spatial discretization of the fluid phase. These values may then be used in the next step of the overall scheme, namely in solving the VANS equations.

An important distinguishing factor of this step in the FPM is the fact that we neither have a mesh available nor do we consider mass particles. Thus, there is no inherent definition of volume associated to elements of our discretization. This matter necessitates a thorough discussion in order to avoid any unsubstantiated adoption of existing averaging techniques, which would not take our specific numerical framework into account.

While we will continue referring to this task as *coarse graining*, we want to point out that this term is ambiguous in the CFD-DEM literature. In addition to denoting the averaging scheme, it is also used for methods which subsume multiple particles into one larger particle with adapted material and contact model parameters [20, 63, 128, 129]. The purpose of such approaches is the reduction of computational cost in CFD-DEM simulations with high particle count.

Before we transition to an averaging scheme on the discretized fluid phase, we first want to discuss a few aspects which have been omitted when we introduced averages in the derivation of volume averaged conservation laws.

4.2.1 Conservation conditions

In Section 3.2 we have already seen that there are a number of averaging formalisms which are used to derive similar versions of the volume averaged Navier-Stokes equations. While the coarse graining is therefore not limited to the weighted averaging introduced earlier, we will, for the sake of consistency with our derivations, first restrict ourselves to such a kernel based averaging procedure. We recall that the solid phase intrinsic volume averages for a set of particles occupying volumes $\mathcal{V}_{\mathrm{s},j} \subset \mathbb{R}^d$ for $j \in \{1, \dots, N_\mathrm{s}\}$ are given by

$$\varepsilon_\mathrm{s}(\boldsymbol{x}) = \sum_{j=1}^{N_\mathrm{s}} \int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x} - \boldsymbol{y}) \, \mathrm{d}V_y \tag{4.2.1}$$

$$\varepsilon_\mathrm{s}(\boldsymbol{x}) \langle \psi \rangle^\mathrm{s}(\boldsymbol{x}) = \sum_{j=1}^{N_\mathrm{s}} \int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x} - \boldsymbol{y}) \psi(\boldsymbol{y}) \, \mathrm{d}V_y \tag{4.2.2}$$

while particle phase averages are defined as

$$n_\mathrm{p}(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{s}} w(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},i}) \tag{4.2.3}$$

$$n_\mathrm{p}(\boldsymbol{x}) \langle \psi \rangle^\mathrm{p}(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{s}} w(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},i}) \psi_i \tag{4.2.4}$$

As before, $\psi(\boldsymbol{x})$ represents any of the quantities which vary within the solid phase volume and ψ_i denotes quantities with a single value per particle. Note that we consider a fixed point in time for the averaging procedure, so that any time dependencies can be omitted to improve readability.

In Section 3.2 we assumed that $\boldsymbol{x} \in \mathcal{V} \subset \mathcal{V}_\infty$ with all points in $\mathcal{V}$ being "far away" (relative to the kernel length scale) from the domain boundary $\partial\mathcal{V}_\infty$. While this helped to keep the

derivation of conservation laws in terms of averaged quantities simple, we cannot generally make this assumption for the solid phase averages in our scheme. Clearly, particles can get close to walls in many real scenarios, in which case part of the averaging kernel, which is centered in the particle, is outside the computational domain. Consequently, without adapting the kernel shape, the normalization condition (3.2.21) is violated. To better understand the implications of this, we integrate the volume and particle phase averaged fields over the full problem domain and observe that we obtain conservation conditions

$$\int_{\mathcal{V}_\infty} \varepsilon_\mathrm{s}(\boldsymbol{x})\,\mathrm{d}V_x = \sum_{j=1}^{N_\mathrm{s}} V_{\mathrm{s},j} \tag{4.2.5}$$

$$\int_{\mathcal{V}_\infty} \varepsilon_\mathrm{s}(\boldsymbol{x})\langle\psi\rangle^\mathrm{s}(\boldsymbol{x})\,\mathrm{d}V_x = \sum_{j=1}^{N_\mathrm{s}} \int_{\mathcal{V}_{\mathrm{s},j}} \psi(\boldsymbol{y})\,\mathrm{d}V_y \tag{4.2.6}$$

$$\int_{\mathcal{V}_\infty} n_\mathrm{p}(\boldsymbol{x})\,\mathrm{d}V_x = N_\mathrm{s} \tag{4.2.7}$$

$$\int_{\mathcal{V}_\infty} n_\mathrm{p}(\boldsymbol{x})\langle\psi\rangle^\mathrm{p}(\boldsymbol{x})\,\mathrm{d}V_x = \sum_{j=1}^{N_\mathrm{s}} \psi_j \tag{4.2.8}$$

which are fulfilled as long as

$$\int_{\mathcal{V}_\infty} w(\boldsymbol{x}_{\mathrm{s},i} - \boldsymbol{y})\,\mathrm{d}V_y = 1 \quad \forall\, i \in \{1, \ldots, N_\mathrm{s}\} \tag{4.2.9}$$

The first equation (4.2.5) states that ε_s should represent the amount of solid volume in the computational domain. Physically reasonable values for this field are of particular importance, since it is used in the calculation of drag forces and occurs in both the volume averaged equations (3.2.95) and (3.2.96) of the fluid phase. A field which does not satisfy the above criterion can certainly be considered unphysical. Clearly, (4.2.7) has a similar significance. Notably, for monodisperse assemblies, which are the only ones considered within this work, this condition is identical to (4.2.5). Another example which illustrates the importance of the above conditions is obtained by recalling that $\boldsymbol{f}_\mathrm{fs}$ can be written as $\boldsymbol{f}_\mathrm{fs} = n_\mathrm{p}\langle\boldsymbol{F}_\mathrm{fs}\rangle^\mathrm{p}$. Thus, by replacing ψ in (4.2.8) with the components of $\boldsymbol{F}_\mathrm{fs}$, we observe that the normalization ensures Newton's third law in the form

$$\int_{\mathcal{V}_\infty} \boldsymbol{f}_\mathrm{fs}(\boldsymbol{x})\,\mathrm{d}V_x = \sum_{j=1}^{N_\mathrm{s}} \boldsymbol{F}_{\mathrm{fs},i} \tag{4.2.10}$$

The significance of this requirement in a coupled scheme is evident, since it ensures that the total force applied by the solid particles on the fluid phase volume (via the source term $\boldsymbol{f}_\mathrm{fs}$ on the right-hand side of (3.2.96)) equals the sum of forces due to the fluid phase on all immersed particles in the solid phase, albeit with reversed sign. Lastly, to illustrate the meaning of (4.2.6) we choose ψ to be the solid continuum velocity which is detailed in Appendix D and can be written as

$$\boldsymbol{u}_\mathrm{s}(\boldsymbol{y}) = \boldsymbol{v}_{\mathrm{s},i} + \boldsymbol{\omega}_{\mathrm{s},i} \times (\boldsymbol{y} - \boldsymbol{x}_{\mathrm{s},i}) \tag{4.2.11}$$

with $\boldsymbol{\omega}_{\mathrm{s},i}$ representing the angular velocity of the particle in a global coordinate system. Since $\boldsymbol{x}_{\mathrm{s},i}$ represents the particle centroid position, it is easy to verify that integrating the second term in (4.2.11) over the particle volume leads to zero, which in turn reduces (4.2.6) to

$$\int_{\mathcal{V}_\infty} \varepsilon_\mathrm{s}(\boldsymbol{x})\langle\boldsymbol{u}_\mathrm{s}\rangle^\mathrm{s}(\boldsymbol{x})\,\mathrm{d}V_x = \sum_{j=1}^{N_\mathrm{s}} \boldsymbol{v}_{\mathrm{s},i} V_{\mathrm{s},i} \tag{4.2.12}$$

Thus, as long as the kernel normalization is not violated, the solid-phase intrinsic average velocity fulfills the momentum conservation condition (4.2.12). Similarly, if we take the particle phase average of the translational solid velocity, (4.2.8) leads to

$$\int_{\mathcal{V}_\infty} n_\mathrm{p}(\boldsymbol{x})\langle \boldsymbol{v}_\mathrm{s}\rangle^\mathrm{p}(\boldsymbol{x})\,\mathrm{d}V_x = \sum_{j=1}^{N_\mathrm{s}} \boldsymbol{v}_{\mathrm{s},i} \tag{4.2.13}$$

With respect to a coupled scheme, averaged solid velocities are only of importance if some of the velocity-dependent terms within $\boldsymbol{F}_\mathrm{fs}$ are evaluated directly in the fluid solver instead of subsuming them into $\boldsymbol{f}_\mathrm{fs}$. An example of this will be given in Section 4.3 where we consider the implicit treatment of drag within the fluid phase.

From the above discussion it is evident that a normalization of the averaging kernel on the computational domain ensures the conservation of physically important properties throughout the averaging procedure and therefore needs to be maintained close to boundaries. In order to commence with boundary treatment, we first discuss approaches to alter the weighting function in such a way that normalization is recovered on the continuous level. This part is independent of the chosen numerical scheme for the fluid phase. After that, we concern ourselves with the values assigned to numerical points in our generalized finite difference framework.

4.2.2 Kernel mirroring

As first step in transitioning from unbounded domains to bounded ones, let us consider half-spaces in $\mathbb{R}^d$. A theoretically elegant approach to restoring the normalization on these domains has been proposed in [176]. For this, we represent the hyperplane which bounds the half-space, by a point $\boldsymbol{x}_\mathrm{w} \in \mathbb{R}^d$ on the hyperplane and the inward pointing unit normal $\boldsymbol{n}_\mathrm{w}$. Then, the computational domain is given by the open set

$$\mathcal{V}_\infty \coloneqq \left\{\boldsymbol{x} \in \mathbb{R}^d | (\boldsymbol{x} - \boldsymbol{x}_\mathrm{w}) \cdot \boldsymbol{n}_\mathrm{w} > 0\right\} \tag{4.2.14}$$

and we define

$$\mathcal{M}_\mathrm{w} : \mathcal{V}_\infty \to \mathcal{V}_\infty^\mathcal{M},\ \boldsymbol{x} \mapsto \boldsymbol{x} - 2[(\boldsymbol{x} - \boldsymbol{x}_\mathrm{w}) \cdot \boldsymbol{n}_\mathrm{w}]\boldsymbol{n}_\mathrm{w} \tag{4.2.15}$$

which maps onto the open set

$$\mathcal{V}_\infty^\mathcal{M} \coloneqq \left\{\boldsymbol{x} \in \mathbb{R}^d | (\boldsymbol{x} - \boldsymbol{x}_\mathrm{w}) \cdot \boldsymbol{n}_\mathrm{w} < 0\right\} \tag{4.2.16}$$

This bijective mapping represents a mirroring across the wall (see Figure 4.7).

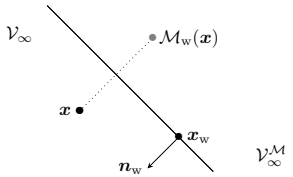


Figure 4.7: Visualization of mirroring (4.2.15) at a single hyperplane.

As stated in [176], though without proof, one can use this mapping to modify a weighting function which is normalized on $\mathbb{R}^d$ in order to achieve a normalization on $\mathcal{V}_\infty$. Specifically, we show the following identity:

Theorem 4.1 *Let $\mathcal{V}_\infty$ and $\mathcal{V}_\infty^{\mathcal{M}}$ be defined as above and $w(\boldsymbol{x}) = \tilde{w}(\|\boldsymbol{x}\|)$ with a bounded and integrable kernel $\tilde{w} : \mathbb{R}_+ \to \mathbb{R}_+$ which satisfies*

$$\int_{\mathbb{R}^d} w(\boldsymbol{y} - \boldsymbol{x}) \, \mathrm{d}V_y = 1, \quad \forall \boldsymbol{x} \in \mathcal{V}_\infty \tag{4.2.17}$$

Then, with the above mapping $\mathcal{M}_\mathrm{w}$ the identity

$$\int_{\mathcal{V}_\infty} w(\boldsymbol{y} - \boldsymbol{x}) + w(\boldsymbol{y} - \mathcal{M}_\mathrm{w}(\boldsymbol{x})) \, \mathrm{d}V_y = 1, \quad \forall \boldsymbol{x} \in \mathcal{V}_\infty \tag{4.2.18}$$

holds true.

Proof: Firstly, we decompose the integral over $\mathbb{R}^d$ to obtain

$$1 = \int_{\mathcal{V}_\infty} w(\boldsymbol{y} - \boldsymbol{x}) \, \mathrm{d}V_y + \int_{\mathcal{V}_\infty^{\mathcal{M}}} w(\boldsymbol{y} - \boldsymbol{x}) \, \mathrm{d}V_y + \int_{\partial\mathcal{V}_\infty} w(\boldsymbol{y} - \boldsymbol{x}) \, \mathrm{d}V_y \tag{4.2.19}$$

We observe that in the last integral, $\partial\mathcal{V}_\infty$ represents the hyperplane. Since we assumed $\tilde{w}$ to be bounded, the integral over this set of measure zero vanishes. Next, let us consider the kernel integral over the image set $\mathcal{V}_\infty^{\mathcal{M}}$. Firstly, by applying the change of variable formula we obtain

$$\int_{\mathcal{V}_\infty^{\mathcal{M}}} w(\boldsymbol{y} - \boldsymbol{x}) \, \mathrm{d}V_y = \int_{\mathcal{V}_\infty} w(\mathcal{M}_\mathrm{w}(\boldsymbol{y}) - \boldsymbol{x}) \left| \det\left(\boldsymbol{\nabla} \mathcal{M}_\mathrm{w}^T \right) \right| \mathrm{d}V_y \tag{4.2.20}$$

For the gradient of the mapping $\mathcal{M}_\mathrm{w}$ one can easily derive that

$$\boldsymbol{\nabla} \mathcal{M}_\mathrm{w} = \mathbb{1} - 2\boldsymbol{n}_\mathrm{w} \otimes \boldsymbol{n}_\mathrm{w} \tag{4.2.21}$$

Using the general identity $\det(a\mathbb{1} + \boldsymbol{b} \otimes \boldsymbol{c}) = a^2(a + \boldsymbol{b} \cdot \boldsymbol{c})$ and the fact that $\|\boldsymbol{n}_\mathrm{w}\| = 1$, provides us with the first equality in

$$\int_{\mathcal{V}_\infty^{\mathcal{M}}} w(\boldsymbol{y} - \boldsymbol{x}) \, \mathrm{d}V_y = \int_{\mathcal{V}_\infty} w(\mathcal{M}_\mathrm{w}(\boldsymbol{y}) - \boldsymbol{x}) \, \mathrm{d}V_y = \int_{\mathcal{V}_\infty} w(\boldsymbol{y} - \mathcal{M}_\mathrm{w}(\boldsymbol{x})) \, \mathrm{d}V_y \tag{4.2.22}$$

while the second equality is a consequence of $w(\boldsymbol{x}) = \tilde{w}(\|\boldsymbol{x}\|)$ and

$$\|\mathcal{M}_\mathrm{w}(\boldsymbol{y}) - \boldsymbol{x}\| = \|\boldsymbol{y} - \mathcal{M}_\mathrm{w}(\boldsymbol{x})\| \tag{4.2.23}$$

which is easily verified by writing out both norms. Thus, by using (4.2.22) in (4.2.19), the final identity is shown. □

From this identity we can conclude that, by replacing the weights in (4.2.1)-(4.2.4) by the sums

$$w(\boldsymbol{x} - \boldsymbol{y}) + w(\boldsymbol{x} - \mathcal{M}_\mathrm{w}(\boldsymbol{y})) \tag{4.2.24}$$

$$w(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},i}) + w(\boldsymbol{x} - \mathcal{M}_\mathrm{w}(\boldsymbol{x}_{\mathrm{s},i})) \tag{4.2.25}$$

the coarse graining conservation conditions within the solid phase, i.e. (4.2.5)-(4.2.8), can be ensured even on a bounded domain. In the context of electrostatic potentials (cf. [71]) and heat conduction this method is also employed and often referred to as *method of images* and *method of image charges*, respectively. Especially the connection to heat conduction, which explained in more detail in Appendix F, can be of interest in the context of averaging

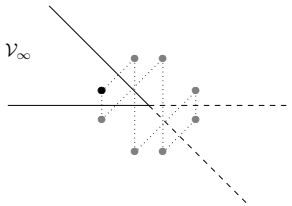


Figure 4.8: Successive mirroring close to a 45 degree corner in a domain bounded by two hyperplanes. The original point from $\mathcal{V}_\infty$ is depicted as a black dot. Mirrored points are depicted as gray dots and dotted lines clarify at which hyperplane points are mirrored.

procedures. Here we only want to state that calculating the solid phase intrinsic and particle phase averages with a normalized Gaussian weighting function and the above mirroring at the boundary leads to the solution of the heat equation

$$\frac{\partial \phi}{\partial \tau} - D\nabla^2 \phi = 0 \tag{4.2.26}$$

with Neumann boundary conditions and evaluated at the (pseudo-)time $T = b^2/4D$. The initial conditions to obtain this result are further given by

$$\phi(\tau = 0, \boldsymbol{x}) = \sum_{j=1}^{N_\mathrm{s}} \psi(\boldsymbol{x}) \chi_{\mathcal{V}_{\mathrm{s},j}}(\boldsymbol{x}) \tag{4.2.27}$$

which leads to $\phi(T, \cdot) = \varepsilon_\mathrm{s} \langle \psi \rangle^\mathrm{s}$ and

$$\phi(\tau = 0, \boldsymbol{x}) = \sum_{j=1}^{N_\mathrm{s}} \psi_j \delta(\|\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},j}\|) \tag{4.2.28}$$

which leads to $\phi(T, \cdot) = n_\mathrm{p} \langle \psi \rangle^\mathrm{p}$. In [14, 149] this equivalence is utilized by realizing the coarse graining procedure through a numerical solution of the heat equation. We will get back to this approach at a later point in this section. For now we want to consider extensions of the mirroring approach to more complex domains.

As is described in Sommerfeld's book [140], one can generalize the procedure to construct fundamental solutions from the half-space bounded by a single hyperplane to some domains bounded by multiple hyperplanes. According to the author it is necessary for the method to be applicable, that the whole space $\mathbb{R}^d$ can be covered completely and simply by successive reflections of the original region at the hyperplanes. In Figure 4.8 we have visualized the process of successive mirroring for a domain in two spatial dimensions which is bounded by two half-lines which extend from a common corner point. It is easily verified that each mirrored point corresponds to an image set of $\mathcal{V}_\infty$, so that $\mathbb{R}^2$ is the disjoint union of the original set, the image sets and sets of measure zero. Thus, $\mathcal{V}_\infty$ in Figure 4.8 is an admissible domain. In particular, every domain bounded by hyperplanes with an angle of π/n, $n \in \mathbb{N}_{>0}$ between the hyperplanes is admissible. A good overview of such domains is provided by Keller [76]. In particular, it turns out that while curved or non-convex boundary can not be treated by the method of images, cuboid geometries form an admissible domain. We make extended use of this fact in our numerical simulations where the mirroring is our method of choice. To provide more detail on why we use this approach, we now want to make the transition to a discretized setting and specify which values are assigned to individual fluid points.

4.2.3 Discretization of averaged fields in GFDM

Due to the finite difference nature of our scheme, the consistent way of assigning values to points is a direct evaluation of the constructed continuous fields at fluid point locations $\boldsymbol{x}_{\mathrm{f},i}$, i.e.

$$\varepsilon_{\mathrm{s},i} = \varepsilon_{\mathrm{s}}(\boldsymbol{x} = \boldsymbol{x}_{\mathrm{f},i}) \tag{4.2.29}$$

$$\varepsilon_{\mathrm{s},i}\langle\psi\rangle_i^{\mathrm{s}} = (\varepsilon_{\mathrm{s}}\langle\psi\rangle^{\mathrm{s}})(\boldsymbol{x} = \boldsymbol{x}_{\mathrm{f},i}) \tag{4.2.30}$$

$$n_{\mathrm{p},i} = n_{\mathrm{p}}(\boldsymbol{x} = \boldsymbol{x}_{\mathrm{f},i}) \tag{4.2.31}$$

$$n_{\mathrm{p},i}\langle\psi\rangle_i^{\mathrm{p}} = (n_{\mathrm{p}}\langle\psi\rangle^{\mathrm{p}})(\boldsymbol{x} = \boldsymbol{x}_{\mathrm{f},i}) \tag{4.2.32}$$

Assuming we have a computational domain which allows for an application of the mirroring approach, we can directly write the number density and particle phase averages as

$$n_{\mathrm{p},i} = \sum_{j=1}^{N_{\mathrm{s}}} w_{ij}, \qquad n_{\mathrm{p},i}\langle\psi\rangle_i^{\mathrm{p}} = \sum_{j=1}^{N_{\mathrm{s}}} w_{ij}\psi_j \tag{4.2.33}$$

where

$$w_{ij} = w(\boldsymbol{x}_{\mathrm{f},i} - \boldsymbol{x}_{\mathrm{s},j}) + \sum_k w(\boldsymbol{x}_{\mathrm{f},i} - \mathcal{M}_{\mathrm{w}}^k(\boldsymbol{x}_{\mathrm{s},j})) \tag{4.2.34}$$

and we have denoted by $\mathcal{M}_{\mathrm{w}}^k$ the sequence of successive mirroring functions as visualized in Figure 4.8. In particular, the force density at fluid phase points is calculated as

$$\boldsymbol{f}_{\mathrm{fs},i} = \sum_{j=1}^{N_{\mathrm{s}}} w_{ij}\boldsymbol{F}_{\mathrm{fs},j} \tag{4.2.35}$$

Thus, by following this procedure, we first ensure that the analytical field $\boldsymbol{f}_{\mathrm{fs}}$ fulfills Newton's third law in the sense of (4.2.10) and then use a discretization of this field which is consistent with the finite difference approach of the FPM. Clearly, we will apply the same line of thought to the volume fraction and volume averages. In these cases, the above evaluation at fluid point positions leads to

$$\varepsilon_{\mathrm{s},i} = \sum_{j=1}^{N_{\mathrm{s}}} \left[\int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x}_{\mathrm{f},i} - \boldsymbol{y})\,\mathrm{d}V_y + \sum_k \int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x}_{\mathrm{f},i} - \mathcal{M}_{\mathrm{w}}^k(\boldsymbol{y}))\,\mathrm{d}V_y \right] \tag{4.2.36}$$

$$\varepsilon_{\mathrm{s},i}\langle\psi\rangle_i^{\mathrm{s}} = \sum_{j=1}^{N_{\mathrm{s}}} \left[\int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x}_{\mathrm{f},i} - \boldsymbol{y})\psi(\boldsymbol{y})\,\mathrm{d}V_y + \sum_k \int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x}_{\mathrm{f},i} - \mathcal{M}_{\mathrm{w}}^k(\boldsymbol{y}))\psi(\boldsymbol{y})\,\mathrm{d}V_y \right] \tag{4.2.37}$$

In practice we certainly do not want to resolve the integral terms in the first two equations via numerical integration. Instead, we assume that

$$\int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x}_{\mathrm{f},i} - \boldsymbol{y})\,\mathrm{d}V_y \approx w(\boldsymbol{x}_{\mathrm{f},i} - \boldsymbol{x}_{\mathrm{s},j})V_{\mathrm{s},j} \tag{4.2.38}$$

$$\int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x}_{\mathrm{f},i} - \mathcal{M}_{\mathrm{w}}^k(\boldsymbol{y}))\,\mathrm{d}V_y \approx w(\boldsymbol{x}_{\mathrm{f},i} - \mathcal{M}_{\mathrm{w}}^k(\boldsymbol{x}_{\mathrm{s},j}))V_{\mathrm{s},j} \tag{4.2.39}$$

which is in line with an assumption we frequently employed in the VANS derivation, namely that the kernel varies little within individual particle volumes. In Appendix G we further verified numerically that the above integral approximation leads to insignificant errors as long as the kernel bandwidth is not too close to the particle radius. Thus, assuming that the bandwidth is chosen large enough, we can write

$$\varepsilon_{\mathrm{s},i} \approx \sum_{j=1}^{N_\mathrm{s}} w_{ij} V_{\mathrm{s},j} \tag{4.2.40}$$

For the definition of solid phase intrinsic averages we only note that, within this work, there is no quantity of interest other than the average velocity, which we assumed to be purely translational in our DEM simulations. Thus, from $\boldsymbol{u}_\mathrm{s}(\boldsymbol{x}) = \boldsymbol{v}_{\mathrm{s},j}$ for $\boldsymbol{x} \in \mathcal{V}_{\mathrm{s},j}$, we directly obtain

$$\varepsilon_{\mathrm{s},i} \langle \boldsymbol{u}_\mathrm{s} \rangle_i^\mathrm{s} \approx \sum_{j=1}^{N_\mathrm{s}} w_{ij} \boldsymbol{v}_{\mathrm{s},j} V_{\mathrm{s},j} \tag{4.2.41}$$

and omit a discussion on the more general case for the sake of brevity. So as long as the computational domains is of a shape that is admissible for the mirroring, a coarse graining based on (4.2.33), (4.2.35), (4.2.40) and (4.2.41) is applicable within the FPM. In the context of this work, we will use this approach exclusively in conjunction with the Gaussian kernel

$$w_b(\boldsymbol{x}) = \frac{1}{(\pi b^2)^{\frac{d}{2}}} e^{-\frac{\|\boldsymbol{x}\|^2}{b^2}} \tag{4.2.42}$$

which is normalized on $\mathbb{R}^d$. Clearly, for the sake of computational efficiency, this kernel is cut off at a certain distance from the origin in a numerical simulation. This distance is typically chosen as a multiple of the kernel bandwidth. This means that, for each particle, only those boundaries which are closer than the cut-off distance of the kernel need to be considered in the mirroring procedure. For all computational domains considered within this work, this further implies that the mirroring procedure reduces to the cases depicted in Figure 4.7 and Figure 4.8 as well as the case of a corner formed by three planes.

While this is sufficient for all numerical simulations considered within the context of this work, we still owe the reader an explanation of our choice and in turn want to put it into context of other approaches that are prevalent in CFD-DEM literature. We will explain which ones can not be properly transferred to our generalized finite difference setting and which of those we discarded due to other disadvantages. The following discussion also provides starting points for possible future extensions to those cases which can not be appropriately treated by the mirroring approach.

4.2.4 Discussion of other approaches

We start with those methods which are typically applied when the fluid phase is numerically solved with the finite volume method (FVM). This is motivated by the fact that a significant portion of the CFD-DEM literature is devoted to this approach. The consistent way of specifying the value ϕ_i attached to a numerical cell $\Omega_i \subset \mathbb{R}^d$ for an analytically provided integrable function $\phi : \mathbb{R}^d \to \mathbb{R}$ in a FVM is

$$\phi_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} \phi(\boldsymbol{x}) \, \mathrm{d}V_x \,, \quad i \in \{1, \ldots, N_\mathrm{f}\} \tag{4.2.43}$$

From this it is immediately clear that, if the analytical fields fulfill the conservation conditions (4.2.5), (4.2.7), (4.2.10) on the domain $\Omega \subset \mathbb{R}^d$, discretized by the cells $\Omega_i \subset \mathbb{R}^d$, the values attached to these cells fulfill the discrete versions

$$\sum_{i=1}^{N_\mathrm{f}} \varepsilon_{\mathrm{s},i} |\Omega_i| = \sum_{j=1}^{N_\mathrm{s}} V_{\mathrm{s},j} \tag{4.2.44}$$

$$\sum_{i=1}^{N_\mathrm{f}} n_{\mathrm{p},i}|\Omega_i| = N_\mathrm{s} \tag{4.2.45}$$

$$\sum_{i=1}^{N_\mathrm{f}} \boldsymbol{f}_{\mathrm{fs},i}|\Omega_i| = \sum_{j=1}^{N_\mathrm{s}} \boldsymbol{F}_{\mathrm{fs},j} \tag{4.2.46}$$

Additionally, if the volume and particle averages are defined as

$$\langle\psi\rangle_i^\mathrm{s} = \frac{1}{\varepsilon_{\mathrm{s},i}|\Omega_i|}\int_{\Omega_i} \varepsilon_\mathrm{s}\langle\psi\rangle^\mathrm{s}(\boldsymbol{x})\,\mathrm{d}V_x\,, \quad i \in \{1,\dots,N_\mathrm{f}\} \tag{4.2.47}$$

$$\langle\psi\rangle_i^\mathrm{p} = \frac{1}{n_{\mathrm{p},i}|\Omega_i|}\int_{\Omega_i} n_\mathrm{p}\langle\psi\rangle^\mathrm{p}(\boldsymbol{x})\,\mathrm{d}V_x\,, \quad i \in \{1,\dots,N_\mathrm{f}\} \tag{4.2.48}$$

analogous discrete versions of (4.2.12) and (4.2.13) are fulfilled. In fact, it is customary in the finite volume context to construct coarse graining schemes specifically so that the averaged values on the fluid grid fulfill discrete conservation conditions (cf. [148, 149]). A very simple approach to achieve this together with a weighted averaging is given by the discrete rescaling

$$w_{ij} = \frac{w(\boldsymbol{x}_{\mathrm{f},i} - \boldsymbol{x}_{\mathrm{s},j})}{\sum\limits_{k=1}^{N_\mathrm{f}} |\Omega_k| w(\boldsymbol{x}_{\mathrm{f},k} - \boldsymbol{x}_{\mathrm{s},j})} \tag{4.2.49}$$

which is very similar to a Shepard type interpolation procedure. This type of discrete kernel rescaling is also frequently applied in the context of smoothed particle hydrodynamics (SPH) [60, 100, 124, 151], which, as we saw earlier, is another mesh free method. In SPH the cell volume is instead replaced by the volume of a SPH particle, which is implicitly given due to the fact that each SPH particle has a certain mass and an attached value of fluid density. We recall from Chapter 2 that an important property of our method is that we don't rely on such mass particles, which would prohibit particle addition and deletion in the presented way. Consequently, there is no inherent definition of point volume which we can use to naturally transfer (4.2.49) to our framework. Furthermore, the above discrete conservation conditions lack a similar theoretical foundation in our framework such that it is not clear whether constructing methods with the sole purpose of fulfilling them is a reasonable approach.

However, if we were to follow this approach, we can make use of the fact that a notion of point volume is available in our framework for the purpose of point management and post-processing. In 2D this "volume", i.e. area, is calculated from a local Delauney triangulation. The center point is assigned one third of the area of each triangle it is a part of (see Figure 4.9). In 3D tetrahedrons are used instead and the point is assigned one fourth of their volume. These triangulations or tetrahedralizations are local in the sense that they are constructed only within the neighborhood of each point and do not necessarily stitch together to form a global mesh. Thus, it is important to realize that this does not contradict the meshfree nature of our numerical scheme. Using the volumes obtained from this approach together with (4.2.49) certainly leads to different results than the mirroring approach close to boundaries. This can be observed from Figure 4.10 where the modification of kernel shape due to mirroring is evidently different from a simple rescaling. A problem of using this in a coarse graining algorithm is the fact that the volumes will be fluctuating over time as the points move and new local decompositions are calculated. These fluctuations might then transfer into the averaged fields and destabilize the coupled simulation. The same obviously does not hold true for the volumes in a FVM and, as long as the density is not fluctuating, also in SPH since there a particle have a fixed mass. We note that the mirroring approach also does not suffer from this deficiency and preliminary simulations

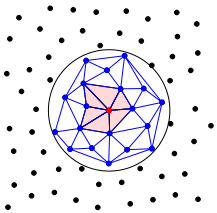


Figure 4.9: Local Delauney triangulation in a FPM point neighborhood in 2D. The red shaded areas contribute to the center point (red point) volume.

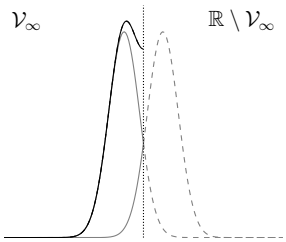


Figure 4.10: Mirroring of Gaussian weighting function at a boundary (dotted) in 1D. The black solid line represents the sum of the original kernel (solid, gray) and mirrored kernel (dashed, gray) inside the domain.

have shown that it provides better stability than the Shepard type correction to the kernel due to the above-mentioned reasons.

Another approach which is fairly common in grid based CFD-DEM is to use the unweighted formalism (see Section 3.2) with grid cells Ω_i as averaging volumes. With this choice and in the above notation, the values attached to cells are obtained by using

$$w_{ij} = \frac{1}{|\Omega_i|} \frac{|\Omega_i \cap \mathcal{V}_{\mathrm{s},j}|}{V_{\mathrm{s},j}} \tag{4.2.50}$$

where $|\mathcal{V}_{\mathrm{s},j}| = V_{\mathrm{s},j}$. Now the implementations of unweighted coarse graining typically differ in the way that the geometrical volumes $|\Omega_i \cap \mathcal{V}_{\mathrm{s},j}|$ are calculated. An exact distribution of each particles volume and quantities to the surrounding cells leads to the so-called *divided particle volume method* (DPVM), sometimes also referred to as *analytical method* [21, 149]. While this might be viable for simple particle and grid cell shapes, this can become computationally infeasible when this is not the case any more or the number of solids is large. Instead a very rough approximation is often used in CFD-DEM simulations when the particle size is much smaller than the grid cell size. This approach, commonly referred to as *particle centroid method* (PCM) results from the approximation

$$|\Omega_i \cap \mathcal{V}_{\mathrm{s},j}| = \chi_{\Omega_i}(\boldsymbol{x}_{\mathrm{s},j})|\mathcal{V}_{\mathrm{s},j}| \tag{4.2.51}$$

It effectively assigns the whole volume of a particle and its attached quantities to the grid cell that the centroid is located in. This method typically leads to little smoothness in time and space, which is detrimental to the stability of coupled simulations. Some methods which improve on PCM, but are still not exact in the sense of DPVM, are summarized for example in [11, 21, 108]. Obviously, the PCM is the computationally most efficient approach, while better approximations towards the DPVM come at the expense of increased time spent on the coarse graining step in coupled simulations. Note that all the above mentioned approaches based on unweighted averaging fulfill the postulated conservation requirements. However, one of the major flaws in many of the DPVM approximations is the fact that one couples the size of averaging volumes to the numerical discretization of the fluid phase. In general, there is no guarantee that the averages defined this way fulfill the assumptions on which the volume averaged conservation laws are based. Usually, in the context of CFD-DEM, this means that an averaging volume needs to be chosen so that it is several times larger than a single particle. Especially in cases where adaptive mesh refinement is necessary, this is

easily violated and the framework can become theoretically questionable. So even if we were to construct similar approaches in our framework, for example based on Voronoi cells from the above local triangulation of the fluid point cloud, we would be confined by the same problems, not shared by the mirroring approach.

In grid-based CFD-DEM methods this problem is typically tackled by applying *two-grid methods*. Therein, a second grid is constructed on a length scale which provides averages that are in agreement with the assumptions inherent in the VANS. The averaged quantities are then calculated on this second grid and interpolated onto the original cells used in computing the fluid flow. Generalizing this approach to our mesh free method would either involve the construction of a background mesh or a second point cloud. Conceptually these two approaches are fairly similar, since they both provide a static set of volumes which exactly or approximately discretize the volume of the computational domain. While the volume definition for cells in a background grid is fixed, one can either use a local Voronoi tessellation, a global one or yet another approach in order to specify volumes on the static point cloud. Clearly, using a second point cloud would be more consistent with our GFDM approach and would allow us to use the standard FPM interpolation procedure. Despite this advantage, it is interesting to see that cylindrical [23, 24, 137] and Cartesian [22, 49] background grids have been applied in fully Lagrangian CFD-DEM schemes based on SPH. The fact that these background meshes only depend on the geometry and not the current state of the fluid, reveals an interesting point that we didn't discuss so far and needs to be considered in mesh free methods, in contrast to mesh-based ones. Namely, if we were to consider free surface flows, our point cloud only discretizes the volume that is currently occupied by the fluid phase. If we take the discrete kernel rescaling (4.2.49) as an example, the formula is very similar in FVM-DEM and our framework. The only difference lies in the fact that the denominator is calculated on the fluid point cloud and with the respective notion of point volume. But the above treatment of free surfaces in our framework reveals a more significant difference. While a rescaling on the fluid point cloud effectively considers the free surface as a boundary to the coarse graining, a rescaling on the grid cells, which are not fitted to the surface, only consider the bounding geometry as coarse graining boundary. At this point we also recall the connection between the mirroring approach and the heat equation. In the context of grid based CFD-DEM the solution of a heat equation, as initially proposed by Capecelatro et al. [14] as well as Sun and Xiao [148, 149], is a prominent method to coarse graining. On simple geometries this should reproduce the result of the mirroring approach, but we observe that this method will also behave slightly different within a mesh free framework. Since the heat equation necessarily has to be solved on the fluid point cloud, the free surface again acts as a coarse graining boundary. On the one hand, this would be alleviated by a background point cloud, which would also provide a volume definition without the previously discussed fluctuations. On the other hand, this becomes very memory intensive when flow domains are large compared to the volume that is actually occupied by fluid at any given time during the simulation.

All these point exemplify that adopting averaging techniques from a mesh-based framework into a Lagrangian GFDM is not a straight-forward task and needs to be handled with care. For the numerical simulations within the present work, we chose to consider problem domains which are fully discretized by the fluid point cloud at any point in time, such that the above pitfalls do not exist. This allows for an evaluation of simulation results with confidence in the foundation of the averaging approach. The above considerations do however form an interesting starting point for further discussion and analysis in future extensions of the presented work.

4.3 Implicit drag calculation

If we strictly follow the procedure presented in the preceding section, we first calculate all coupling forces acting on the solid particles and then average these values in a coarse graining procedure to obtain the force density $\boldsymbol{f}_{\mathrm{fs}}$. This implies that any fluid phase quantity which is used in the calculation of $\boldsymbol{f}_{\mathrm{fs}}$ corresponds to the previous time level in our coupled algorithm (cf. Figure 4.1). However, in Section 3.4 we have seen that the drag force has the form

$$\boldsymbol{F}_{\mathrm{d}} = \frac{V_{\mathrm{s}}\beta}{\varepsilon_{\mathrm{s}}}(\boldsymbol{u}_{\mathrm{f}} - \boldsymbol{v}_{\mathrm{s}}) \tag{4.3.1}$$

with a coefficient β which itself depends on both, the fluid volume fraction and relative velocity, i.e.

$$\beta = \beta(\varepsilon_{\mathrm{f}}, \|\boldsymbol{u}_{\mathrm{f}} - \boldsymbol{v}_{\mathrm{s}}\|) \tag{4.3.2}$$

Clearly, the dependency of $\boldsymbol{F}_{\mathrm{d}}$ on the fluid velocity also translate into the source term

$$\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{s}}} w(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},i})\boldsymbol{F}_{\mathrm{d},i} \tag{4.3.3}$$

such that an explicit treatment of drag in time integration schemes for both, the solid and fluid momentum equation, imposes a time step restriction. For the solid phase we already discussed the fact that we only consider explicit time integration schemes. A time step restriction is therefore unavoidable and will be discussed later in this chapter. On the other hand, for the fluid momentum equation we use an implicit time integration scheme such that it suggests itself to investigate the possibility of treating the drag term implicitly. To see how this can be achieved, we plug (4.3.1) into (4.3.3) to obtain

$$\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}) = \sum_{i=1}^{N_{\mathrm{s}}} w(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},i})\frac{\beta_i}{\varepsilon_{\mathrm{s},i}}(\boldsymbol{u}_{\mathrm{f},i} - \boldsymbol{v}_{\mathrm{s},i})V_{\mathrm{s},i} \tag{4.3.4}$$

Now, we recall from the derivation of the VANS equations that averaged fields can be assumed to vary little within the kernel range, such that they can be evaluated in $\boldsymbol{x}$ instead of $\boldsymbol{x}_{\mathrm{s},i}$. By applying this to the volume fraction and fluid velocity in (4.3.4) and moving them out of the summation, we obtain the approximation

$$\boldsymbol{f}_{\mathrm{d}}(\boldsymbol{x}) \approx \frac{\sum_{i=1}^{N_{\mathrm{s}}} w(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},i})\beta_i V_{\mathrm{s},i}}{\varepsilon_{\mathrm{s}}(\boldsymbol{x})}\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x}) - \frac{\sum_{i=1}^{N_{\mathrm{s}}} w(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},i})\beta_i \boldsymbol{v}_{\mathrm{s},i} V_{\mathrm{s},i}}{\varepsilon_{\mathrm{s}}(\boldsymbol{x})} \tag{4.3.5}$$

which can also be written as

$$\boldsymbol{f}_{\mathrm{d}} \approx \langle\beta\rangle^{\mathrm{s}}\boldsymbol{u}_{\mathrm{f}} - \langle\beta\boldsymbol{u}_{\mathrm{s}}\rangle^{\mathrm{s}} \tag{4.3.6}$$

We further introduce the notation

$$\boxed{\frac{\boldsymbol{f}_{\mathrm{d}}}{\varepsilon_{\mathrm{f}}\rho_{\mathrm{f}}} = B(\boldsymbol{u}_{\mathrm{f}} - \boldsymbol{u}_B)} \tag{4.3.7}$$

with

$$B = \frac{\langle\beta\rangle^{\mathrm{s}}}{\rho_{\mathrm{f}}\varepsilon_{\mathrm{f}}}, \qquad \boldsymbol{u}_B = \frac{\langle\beta\boldsymbol{u}_{\mathrm{s}}\rangle^{\mathrm{s}}}{\langle\beta\rangle^{\mathrm{s}}} \tag{4.3.8}$$

which will prove to be useful when deriving the numerical scheme for the VANS equations. Alternatively, we can introduce the notation

$$\boldsymbol{F}_{\mathrm{d}} = \alpha(\boldsymbol{u}_{\mathrm{f}} - \boldsymbol{v}_{\mathrm{s}}) \tag{4.3.9}$$

with $\alpha = V_{\mathrm{s}}\beta/\varepsilon_{\mathrm{s}}$ in (4.3.1). With the same approximation as before it is easy to see that the terms in (4.3.8) become

$$B = \frac{n_\mathrm{p}\langle\alpha\rangle^\mathrm{p}}{\rho_\mathrm{f}\varepsilon_\mathrm{f}}, \qquad \boldsymbol{u}_B = \frac{n_\mathrm{p}\langle\alpha\boldsymbol{v}_\mathrm{s}\rangle^\mathrm{p}}{n_\mathrm{p}\langle\alpha\rangle^\mathrm{p}} \tag{4.3.10}$$

which is actually closer to the way that (4.3.7) is treated in our implementation. Now, if we were to consider the simple Stokes drag for an isolated sphere

$$\boldsymbol{F}_\mathrm{d} = 3\pi\mu D_\mathrm{s}\varepsilon_\mathrm{f}(\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}) \tag{4.3.11}$$

we directly see that β and α are independent of the fluid velocity. Thus, when a first order implicit time integration is applied to the momentum equation (3.2.96), we can employ a fully implicit treatment of drag in the sense that

$$\frac{\boldsymbol{u}_\mathrm{f}^{n+1} - \boldsymbol{u}_\mathrm{f}^{n}}{\Delta t} = \ldots - B\Big(\boldsymbol{u}_\mathrm{f}^{n+1} - \boldsymbol{u}_\mathrm{B}\Big) \tag{4.3.12}$$

where B and $\boldsymbol{u}_\mathrm{B}$ are independent of the fluid velocity and can be calculated based on the already available values of $\varepsilon_\mathrm{s}^{n+1}$ and $\boldsymbol{v}_\mathrm{s}^{n+1}$. However, as we have seen from the expressions for C_d, χ and β in Section 3.4, the drag force is typically highly nonlinear in the fluid velocity, which is why such a fully implicit treatment is not possible. Instead, we consider a semi-implicit scheme, where β is calculated within the DEM solver based on the fluid velocity from the previous time step $\boldsymbol{u}_\mathrm{f}^n$ and then averaged to obtain B and $\boldsymbol{u}_\mathrm{B}$ which are considered constant within the time integration (4.3.12).

While (4.3.8) and (4.3.10) are the formulations we employ in the following, there are a variety other ways to incorporate the drag term in the VANS which can be found in CFD-DEM literature. Upon comparison with these sources we note that the above formulation, which somewhat naturally followed from the averaging formalism, is similar to the one used in [168]. In terms of the general classification of coupling schemes proposed by Feng and Yu [48], using this formulation classifies as a scheme of type 3, i.e. one that calculates forces in the solid phase first and then projects back to the fluid phase. Surely, other procedures that either fall in the category of type 1 schemes, i.e. coupling terms are calculated in the same way they would be in a two-fluid scheme, or fall somewhere in between, can also be found in CFD-DEM literature. For example, the authors in [52] employ a type 1 coupling of the form

$$\boldsymbol{f}_\mathrm{d} = \beta(\boldsymbol{u}_\mathrm{f} - \langle\boldsymbol{u}_\mathrm{s}\rangle^\mathrm{s}, \varepsilon_\mathrm{f})(\boldsymbol{u}_\mathrm{f} - \langle\boldsymbol{u}_\mathrm{s}\rangle^\mathrm{s}) \tag{4.3.13}$$

while the approach in [160] would in our notation take the form

$$\boldsymbol{f}_\mathrm{d} = \langle\beta\rangle^\mathrm{s}(\boldsymbol{u}_\mathrm{f} - \langle\boldsymbol{u}_\mathrm{s}\rangle^\mathrm{s}) \tag{4.3.14}$$

In some part of CFD-DEM literature the coupling is also done by considering the forces onto particles and source term in the fluid phase separately through drag correlations on the one hand and porous media models on the other [22]. While this is a more phenomenologically motivated approach, our discussions on drag forces have shown that there is a tight link between the drag correlations and Darcy type laws and it is thus similar to (4.3.13). We note that a coupling similar to (4.3.13) can also be found in the work by Drumm et al. [40] where the authors considered liquid-liquid two-phase flow in the finite pointset method. This will be further discussed in the next section.

An aspect which we left unconsidered so far is the fact that $\boldsymbol{f}_\mathrm{d}$ from (4.3.7) with either (4.3.8) or (4.3.10) and the old fluid velocity is not able to exactly fulfill a condition of the form (4.2.46), even if the coarse graining scheme is constructed to do so. This is due to the approximation made in the above derivations. To address this, while still allowing for the

(semi-)implicit treatment of drag, one can incorporate the function approximation stencils, used in the solid phase, into the calculation of $\boldsymbol{f}_\mathrm{d}$. To be precise, one can write

$$\sum_{j=1}^{N_\mathrm{s}} w_{ij}\alpha_j \boldsymbol{u}_\mathrm{f}(\boldsymbol{x}_{\mathrm{s},j}) = \sum_{j=1}^{N_\mathrm{s}} w_{ij}\alpha_j \left(\sum_{k=1}^{N_\mathrm{f}} c_{jk}^0 \boldsymbol{u}_{\mathrm{f},k} \right) = \sum_{k=1}^{N_\mathrm{f}} \boldsymbol{u}_{\mathrm{f},k} \left(\sum_{j=1}^{N_\mathrm{s}} w_{ij}\alpha_j c_{jk}^0 \right) \tag{4.3.15}$$

with c_{jk}^0 denoting the components of the function value approximation stencil as discussed in Chapter 2. We observe that one can define the term in parenthesis as a new stencil

$$c_{ik}^\alpha = \sum_{j=1}^{N_\mathrm{s}} w_{ij}\alpha_j c_{jk}^0 \tag{4.3.16}$$

for each fluid point $i = 1, \ldots, N_\mathrm{f}$ and write

$$\boldsymbol{f}_{\mathrm{d},i} = \sum_{k=1}^{N_\mathrm{f}} c_{ik}^\alpha \boldsymbol{u}_{\mathrm{f},k} - \sum_{j=1}^{N_\mathrm{s}} w_{ij}\alpha_j \boldsymbol{v}_{\mathrm{s},j} \tag{4.3.17}$$

Clearly, this formulation would still enable us to employ a semi-implicit coupling, the only difference being the fact that in the linear system, the matrix corresponding to velocity components is now filled more densely. Since we did not use the previous approximation, Newton's third law in the form (4.2.46) can now be exactly fulfilled if the coarse graining scheme is build to do so and the old fluid velocity is used.

While all of the above expression from CFD-DEM literature have shown to be applicable, we stick to (4.3.10) which we derived directly from our averaging formalism. An extensive comparison of these approaches is better suited for future work when the CFD-DEM algorithm is firmly established in our numerical framework. In comparison to (4.3.17), expression (4.3.7) further provides us with an easier way of formulating the numerical scheme while only introducing a degree of approximation which can be considered negligible, given the assumptions already inherent in the VANS.

4.4 Numerical scheme for the volume averaged equations

After having introduced both, the DEM and coarse graining scheme, we now want to consider the third, and last, component of every time iteration within the coupled algorithm we sketched in Figure 4.1. Clearly, we refer to the numerical scheme used to solve the volume averaged Navier-Stokes (VANS) equations within our generalized finite difference framework. For this, we adapt the projection scheme, presented in [73] for incompressible flows, to the multiphase setting of this work. Thus, in the following we consider

$$\frac{\mathrm{D}\boldsymbol{x}_\mathrm{f}}{\mathrm{D}t} = \boldsymbol{u}_\mathrm{f} \tag{4.4.1}$$

$$\frac{\mathrm{D}(\varepsilon_\mathrm{f}\rho_\mathrm{f})}{\mathrm{D}t} = -\varepsilon_\mathrm{f}\rho_\mathrm{f}(\boldsymbol{\nabla}\cdot\boldsymbol{u}_\mathrm{f}) \tag{4.4.2}$$

$$\tag{4.4.3}$$

together with either

$$\frac{\mathrm{D}\boldsymbol{u}_\mathrm{f}}{\mathrm{D}t} = -\frac{1}{\varepsilon_\mathrm{f}\rho_\mathrm{f}}\boldsymbol{\nabla}p + \frac{1}{\varepsilon_\mathrm{f}\rho_\mathrm{f}}\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_\mathrm{f} - B(\boldsymbol{u}_\mathrm{f} - \boldsymbol{u}_\mathrm{B}) + \tilde{\boldsymbol{g}} \tag{4.4.4}$$

or

$$\frac{\mathrm{D}\boldsymbol{u}_\mathrm{f}}{\mathrm{D}t} = -\frac{1}{\rho_\mathrm{f}}\boldsymbol{\nabla}p + \frac{1}{\rho_\mathrm{f}}\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_\mathrm{f} - B(\boldsymbol{u}_\mathrm{f} - \boldsymbol{u}_\mathrm{B}) + \tilde{\boldsymbol{g}} \tag{4.4.5}$$

depending on the choice of coupling discussed in Section 3.3. Note that we have rewritten the different models of VANS equations (4.4.4) and (4.4.5) in such a way that both, an implicit and explicit coupling, can be represented. If we were to consider a completely explicit coupling, the drag force $\boldsymbol{f}_\mathrm{d}$ can be subsumed in the modified body force term, i.e.

$$\tilde{\boldsymbol{g}} = \boldsymbol{g} - \frac{1}{\varepsilon_\mathrm{s}\rho_\mathrm{f}} \begin{cases} \boldsymbol{f}_\mathrm{d} + \boldsymbol{f}_\mathrm{p} + \boldsymbol{f}_\tau & \text{for model I} \\ \boldsymbol{f}_\mathrm{d} & \text{for model II} \end{cases} \tag{4.4.6}$$

and the constant B can be set to zero. If we consider the semi-implicit coupling proposed in the previous section, the drag force is not included in the body force term and B as well as $\boldsymbol{u}_\mathrm{B}$ are chosen according to (4.3.10). Note that we have once more omitted the averaging notation to improve readability. As before, we consider the fluid material density to be constant, such that the mass conservation equation reduces to

$$\frac{\mathrm{D}\varepsilon_\mathrm{f}}{\mathrm{D}t} = -\varepsilon_\mathrm{f}(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f}) \tag{4.4.7}$$

Although the volume fraction indirectly depends on fluid velocity and pressure, since they influence the solid motion, an implicit treatment would necessitate the solution of both, the fluid and solid motion, within a single large system of equations. For general contact models and coupling terms this is certainly not feasible, which motivated the proposed sequential procedure of DEM, followed by coarse graining and the integration of VANS equations. For this reason, we assume that, from the perspective of the fluid solver, the volume fraction is a quantity which is known at both the old and new time level. Thus, similar to incompressible flows, the mass conservation equation provides a kinematic constraint on the velocity field, rather than a dynamic equation determining the transport of fluid density. Also, the pressure influences all components of velocity through its gradient, but there is no equation that can be explicitly solved for the pressure. To tackle this problem, we apply a scheme that is based on a projection approach. Projection methods are a subset of fractional step methods which originiated in the scheme propsed by Chorin [17] for the incompressible Navier-stokes equations. In a projection method one first calculates an intermediate velocity field from the momentum equations without requiring that it satisfies the continuity equation, then one restores this porperty through the use of a suitable pressure. The name *projection method* is a result of the incompressible single-phase case, where the correction step can be described as a projection onto the space of divergence-free fields.

To compute the intermediate velocity $\hat{\boldsymbol{u}}_\mathrm{f}$ we consider a first order implicit time discretization of the momentum equations (4.4.4) or (4.4.5). We note that, since our Lagrangian approach avoids the discretization of convective terms, higher order schemes are not necessary. This is further supported by the analysis of such schemes in the context of the current type of projection methods, which are found in [131]. There the authors observed a significant increase in computational cost without a compensating gain in accuracy. To obtain a unified numerical scheme for both models we further introduce the notation

$$\tilde{\rho_\mathrm{f}}^{n+1} = \begin{cases} \varepsilon_\mathrm{f}^{n+1}\rho_\mathrm{f} & \text{for model I} \\ \rho_\mathrm{f} & \text{for model II} \end{cases} \tag{4.4.8}$$

so that

$$\frac{\hat{\boldsymbol{u}}_\mathrm{f}^{n+1} - \boldsymbol{u}_\mathrm{f}^{n}}{\Delta t} = -\frac{1}{\tilde{\rho_\mathrm{f}}^{n+1}}\boldsymbol{\nabla}\tilde{p}^{n+1} + \frac{1}{\tilde{\rho_\mathrm{f}}^{n+1}}\boldsymbol{\nabla} \cdot \boldsymbol{\tau}_\mathrm{f}\left(\hat{\boldsymbol{u}}_\mathrm{f}^{n+1}\right) - B\left(\hat{\boldsymbol{u}}_\mathrm{f}^{n+1} - \boldsymbol{u}_\mathrm{B}\right) + \tilde{\boldsymbol{g}}^{n+1} \tag{4.4.9}$$

We note that superscripts denoting the time level are omitted for B and $\boldsymbol{u}_\mathrm{B}$ to improve readability. It has been clarified in the previous section that they are constant within the fluid solver time step since they are based on fluid quantities at the previous time level and have been calculated within the coarse graining procedure preceding the VANS solution scheme. We have further replaced the unknown pressure p^{n+1} by a guess $\tilde{p}^{n+1}$ which is independent of the intermediate velocity. This value can be chosen in different ways to improve how well the intermediate velocity approximates the target velocity. The most obvious guess is given by the pressure at the previous time level, i.e. $\tilde{p}^{n+1} = p^n$. We instead apply a splitting of the pressure into a hydrostatic and dynamic part

$$p^{n+1} = p_\mathrm{hyd}^{n+1} + p_\mathrm{dyn}^{n+1} \tag{4.4.10}$$

An equation for the hydrostatic pressure is obtained from the momentum equation by keeping only terms that are non-zero for vanishing velocities and then applying the divergence operator to obtain a scalar equation. Thus, from (4.4.9) we get

$$\boldsymbol{\nabla}\cdot\left(\frac{1}{\tilde{\rho}_\mathrm{f}^{n+1}}\boldsymbol{\nabla}p_\mathrm{hyd}^{n+1}\right) = \boldsymbol{\nabla}\cdot(B\boldsymbol{u}_\mathrm{B}) + \boldsymbol{\nabla}\cdot\tilde{\boldsymbol{g}}^{n+1} \tag{4.4.11}$$

Assuming we have calculated the hydrostatic pressure from one the above equation, we use the dynamic pressure from the previous time step to set

$$\tilde{p}^{n+1} = p_\mathrm{hyd}^{n+1} + p_\mathrm{dyn}^{n} \tag{4.4.12}$$

Based on this guess, the intermediate velocity is now calculated from (4.4.9). Since this velocity field does satisfy the kinematic constraint (4.4.7), we assume that there is a correction to the pressure guess, denoted by p_corr^{n+1}, so that

$$\frac{\boldsymbol{u}_\mathrm{f}^{n+1}-\boldsymbol{u}_\mathrm{f}^{n}}{\Delta t} = -\frac{1}{\tilde{\rho}_\mathrm{f}^{n+1}}\boldsymbol{\nabla}\left(\tilde{p}^{n+1}+p_\mathrm{corr}^{n+1}\right)+\frac{1}{\tilde{\rho}_\mathrm{f}^{n+1}}\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_\mathrm{f}\left(\boldsymbol{u}_\mathrm{f}^{n+1}\right)-B\left(\boldsymbol{u}_\mathrm{f}^{n+1}-\boldsymbol{u}_\mathrm{B}\right)+\tilde{\boldsymbol{g}}^{n+1} \tag{4.4.13}$$

and the final velocity $\boldsymbol{u}_\mathrm{f}^{n+1}$ satisfies the time discretized form of (4.4.7), given by

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}_\mathrm{f}^{n+1} = -\frac{1}{\Delta t}\ln\left(\frac{\varepsilon_\mathrm{f}^{n+1}}{\varepsilon_\mathrm{f}^{n}}\right) \tag{4.4.14}$$

In a first step to derive an equation for this correction pressure, we now subtract the equations we used to determine the intermediate velocity (4.4.9) from those we just formulated for the corrected velocity (4.4.13) and apply the divergence operator to the result. These calculations lead to

$$\boldsymbol{\nabla}\cdot\left(\frac{\Delta t}{1+\Delta t B}\frac{1}{\tilde{\rho}_\mathrm{f}^{n+1}}\boldsymbol{\nabla}p_\mathrm{corr}^{n+1}\right) = \boldsymbol{\nabla}\cdot\hat{\boldsymbol{u}}_\mathrm{f}^{n+1} - \boldsymbol{\nabla}\cdot\boldsymbol{u}_\mathrm{f}^{n+1} \tag{4.4.15}$$

where we have further assumed that

$$\boldsymbol{\nabla}\cdot\left(\frac{\Delta t}{1+\Delta t B}\frac{1}{\tilde{\rho}_\mathrm{f}^{n+1}}\left(\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_\mathrm{f}\left(\boldsymbol{u}_\mathrm{f}^{n+1}\right)-\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_\mathrm{f}\left(\hat{\boldsymbol{u}}_\mathrm{f}^{n+1}\right)\right)\right)\approx 0 \tag{4.4.16}$$

This simplification is commonly used in projection schemes and represents one of their drawbacks. It is however only problematic in the case of very low Reynolds number, where this approximation might lead to a loss of accuracy and stability [144]. The simulations done in this work do not fall into this category and the presented scheme remains applicable. Using equation (4.4.14), we can easily calculate all terms on the right-hand side and solve the Poisson equation for the correction pressure. Finally we can correct the intermediate velocity field via

$$\boldsymbol{u}_{\mathrm{f}}^{n+1} = \hat{\boldsymbol{u}}_{\mathrm{f}}^{n+1} - \frac{\Delta t}{1+\Delta t B}\frac{1}{\tilde{\rho}_{\mathrm{f}}^{n+1}}\boldsymbol{\nabla} p_{\mathrm{corr}}^{n+1} \tag{4.4.17}$$

We emphasize that the approximation (4.4.16) has no influence on the fact that the velocity field calculated from (4.4.17) does now fulfill the discretized mass conservation. Instead, it causes the combination of new velocity $\boldsymbol{u}_{\mathrm{f}}^{n+1}$ and the pressure given by $\tilde{p}^{n+1} + p_{\mathrm{corr}}^{n+1}$ to not satisfy the momentum conservation exactly. To obtain a pressure with this property we introduce another pressure solution step. The corresponding equation can be derived by again applying the divergence operator to the momentum equations and reformulating any occurrence of the velocity divergence with the help of the continuity equation. Following this idea and using the unified notation for both models, we obtain the scalar equation

$$\boldsymbol{\nabla}\cdot\left(\frac{\mathrm{D}\boldsymbol{u}_{\mathrm{f}}}{\mathrm{D}t}\right) = -\boldsymbol{\nabla}\cdot\left(\frac{1}{\tilde{\rho}_{\mathrm{f}}}\boldsymbol{\nabla} p\right) + \boldsymbol{\nabla}\cdot\left(\frac{1}{\tilde{\rho}_{\mathrm{f}}}\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_{\mathrm{f}}\right) - \boldsymbol{\nabla}\cdot(B\boldsymbol{u}_{\mathrm{f}}) + \boldsymbol{\nabla}\cdot(B\boldsymbol{u}_{\mathrm{B}}) + \boldsymbol{\nabla}\cdot\tilde{\boldsymbol{g}} \tag{4.4.18}$$

Writing out the material derivative and using the tensor identity

$$\boldsymbol{\nabla}\cdot(\boldsymbol{u}_{\mathrm{f}}\cdot\boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}}) = \boldsymbol{u}_{\mathrm{f}}\cdot\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u}_{\mathrm{f}}) + (\boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}})^{T} : \boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}} \tag{4.4.19}$$

we realize that the left-hand side of (4.4.18) becomes

$$\boldsymbol{\nabla}\cdot\left(\frac{\mathrm{D}\boldsymbol{u}_{\mathrm{f}}}{\mathrm{D}t}\right) = \frac{\mathrm{D}(\boldsymbol{\nabla}\cdot\boldsymbol{u}_{\mathrm{f}})}{\mathrm{D}t} + (\boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}})^{T} : \boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}} \tag{4.4.20}$$

Now we include this in equation (4.4.18), discretize in time and cross out all terms that are already included in the new hydrostatic pressure we calculated from (4.4.11). This leads to the following equation for the new dynamic pressure

$$\boldsymbol{\nabla}\cdot\left(\frac{1}{\tilde{\rho}_{\mathrm{f}}}\boldsymbol{\nabla} p_{\mathrm{dyn}}^{n+1}\right) = -\frac{(\boldsymbol{\nabla}\cdot\boldsymbol{u}_{\mathrm{f}})^{n+1} - (\boldsymbol{\nabla}\cdot\boldsymbol{u}_{\mathrm{f}})^{n}}{\Delta t} + \boldsymbol{\Phi}\left(\boldsymbol{u}_{\mathrm{f}}^{n+1}\right) + \boldsymbol{\Psi}\left(\boldsymbol{u}_{\mathrm{f}}^{n+1}\right) + \boldsymbol{\Theta}\left(\boldsymbol{u}_{\mathrm{f}}^{n+1}\right) \tag{4.4.21}$$

where we have further introduced

$$\boldsymbol{\Psi}(\boldsymbol{u}_{\mathrm{f}}) = \boldsymbol{\nabla}\cdot\left(\frac{1}{\tilde{\rho}_{\mathrm{f}}}\boldsymbol{\nabla}\cdot\boldsymbol{\tau}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{f}})\right) \tag{4.4.22}$$

$$\boldsymbol{\Phi}(\boldsymbol{u}_{\mathrm{f}}) = -(\boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}})^{T} : \boldsymbol{\nabla}\boldsymbol{u}_{\mathrm{f}} \tag{4.4.23}$$

$$\boldsymbol{\Theta}(\boldsymbol{u}_{\mathrm{f}}) = -\boldsymbol{\nabla}\cdot(B\boldsymbol{u}_{\mathrm{f}}) \tag{4.4.24}$$

and used a first oder approximation for the material derivative in (4.4.20). Summarizing all of the above steps, the order of computations in our scheme is the following:

1. Calculate a hydrostatic pressure from (4.4.11)
2. Calculate an intermediate velocity field based on the newly computed hydrostatic pressure and the old dynamic pressure from (4.4.9)
3. Calculate a correction pressure to restore the correct divergence of velocity from (4.4.15)
4. Correct the intermediate velocity using (4.4.17)
5. Calculate a dynamic pressure that is consistent with the new velocity from (4.4.21)

We note that in certain cases, such as the above-mentioned low Reynolds number flows, subsuming steps two and three into the solution of a single large linear system for both, the intermediate velocity and correction pressure, can be advantageous. However, this is not used in the context of this work and we refer the interested reader to [146].

Remark 4.1 *Equation* (4.4.14) *reveals a challenge in devising an averaging procedure on moving point clouds. Scaling the density change in a single-phase setting with the inverse fluid time step provides a bounded value, since the density is calculated from quantities which only change by an amount that is tied to the time step. While, at first glance, the same holds true for the volume fraction, due to the particle movement being tied to the time step, it is not a priori clear that this is still fulfilled by the corresponding averaged field. A violation of this property can result for example from the use of local representative volume approximations in the averaging procedure, which are calculated for each point in the point cloud. One can typically not guarantee that the operations necessary to maintain a point cloud with sufficient quality in the differential operators, i.e. point addition, deletion and merging, do not lead to fluctuations in averaged quantities. On the other hand, the weighted averaging with mirroring at the boundary does not rely on point volumes and eliminates this source of fluctuations in the volume fraction.*

4.5 Comparison to previous work in FPM

As we have already noted in Section 4.3, coupled simulations of a continuous and dispersed fluid phase within the finite pointset method have been considered by Drumm et al. in [40]. Now that we have explained all the essential components of our coupled algorithm, we want to clarify which aspects set apart the work of this thesis from their approach and are thus novel within our framework.

Firstly, we note that the volume averaged equations in their case were slightly different from those within this thesis. While we have resorted to the derivation by Anderson and Jackson, the fluid-fluid equations, presented in the above-mentioned paper, are obtained from the work by Ishii [67, 68], as is discussed in [161]. This has no influence on the mass conservation equation, which is identical in their work, but the momentum equations slightly differ in the way that the averaged viscous stress is modeled.

However, the more important distinguishing factor is due to the way that volume fraction and coupling terms are calculated and used within the continuous phase numerical scheme. In their procedure the porosity and in extension the velocity divergence are not determined from the dispersed phase position and volume. Instead the volume fractions in both phases are advanced in time on the basis of the divergence of this phases velocity from a previous time step and rescaled to fulfill $\varepsilon_\mathrm{s}^{n+1} + \varepsilon_\mathrm{f}^{n+1} = 1$ at the end of each time step. Transferring this into our notation leads to

$$\varepsilon_\mathrm{f}^{n+1} = \frac{\varepsilon_\mathrm{f}^{n}}{1 + \Delta t(\boldsymbol{\nabla} \cdot \boldsymbol{u}_\mathrm{f}^{n})} \tag{4.5.1}$$

$$\varepsilon_\mathrm{s}^{n+1} = \frac{\varepsilon_\mathrm{s}^{n}}{1 + \Delta t(\boldsymbol{\nabla} \cdot \boldsymbol{v}_\mathrm{s}^{n})} \tag{4.5.2}$$

which is the result of an implicit first order time discretization of the mass conservation equation under the assumption of constant divergence of velocity. This differs from the time discretization in (4.4.14) since we have used the analytical solution obtained under the same assumption. Apart from this minor difference, we observe that (4.5.2) requires the construction of differential operators on the point cloud which represents the dispersed phase. While this is acceptable in some cases, such as the one presented by the authors in

[40], we do not include the solid phase in our point cloud management and can therefore not guarantee that any of the previously discussed quality criteria are satisfied for this divergence operator. Now, if we still would have wanted to employ (4.5.1) this would have necessitated the subsequent choice of $\varepsilon_{\mathrm{s}}^{n+1} = 1 - \varepsilon_{\mathrm{f}}^{n+1}$. Since a direct connection between dispersed phase positions and volume fraction is only established within the initial condition (not detailed in [40]) and is then advanced based only on fluid velocity, we expect the volume fraction to be rather sensitive to errors in the velocity and potentially destabilize the coupled simulation. Consequently, we have refrained from using this approach and resorted to the coarse graining based on particle positions, which also provides a more clear connection to the theoretical basis of the volume averaged equations. As previously discussed this also motivated the order of calculation, being solid phase first, fluid phase second.

Another key difference between their coupling approach and the one presented here lies in the way that drag force acts on both phases. On the one hand, while the authors in [40] also employed an implicit treatment of drag in the fluid phase, the drag correlation is evaluated directly at fluid points based on an interpolated solid velocity. While this appears reasonable and is similar to the way that drag is handled in two-fluid models, no derivation based on Newton's third law has been provided and it is not clear how cases are treated where dilute particle configurations lead to decreasing quality of approximation stencils. In both of these aspects we view the presented method as more clear and well-founded. On the other hand, the authors did not formulate the drag correlation in terms of superficial velocity and did not include a voidage function, as is the case in the majority of CFD-DEM literature. According to [108] this is only applicable for particles that can be considered to behave similar to an isolated sphere. Thus, in the classification of multiphase coupling by Crowe [27] this corresponds to a two-way coupled scheme, in contrast to the four-way coupling we strife to model.

With all these connections and differences disclosed, we can summarize that, despite the initial similarity one might see, the work done in [40] differs in a number of key components from what we present here and is not applicable to the setups considered in the next chapter.

4.6 Time step restrictions

Up to this point we only provided expressions for time restrictions which also emerge in single-phase flows. On the one hand, we stated the CFL condition (2.3.3) which bounds the fluid phase time step. On the other hand, we discussed time step restrictions due to the modeling of solid particle contacts with a soft-sphere DEM in Section 4.1.5. However, for a coupled simulation we still need to investigate possible time step bounds due to the coupling force $\boldsymbol{F}_{\mathrm{fs}}$ in the solid phase and momentum source $\boldsymbol{f}_{\mathrm{fs}}$ in the fluid phase. In particular, we still owe the reader more details on time step bounds due to drag force, since we already touched upon this subject in section Section 4.3.

4.6.1 Solid phase time step restriction

Let us first investigate the influence of drag on the solid phase time step. To this end, we consider the momentum equation of a particle moving only under the influence of drag, i.e.

$$m_{\mathrm{s},i} \frac{\mathrm{d}\boldsymbol{v}_{\mathrm{s},i}}{\mathrm{d}t} = \boldsymbol{F}_{\mathrm{d},i} = \alpha_i (\boldsymbol{u}_{\mathrm{f},i} - \boldsymbol{v}_{\mathrm{s},i}) \tag{4.6.1}$$

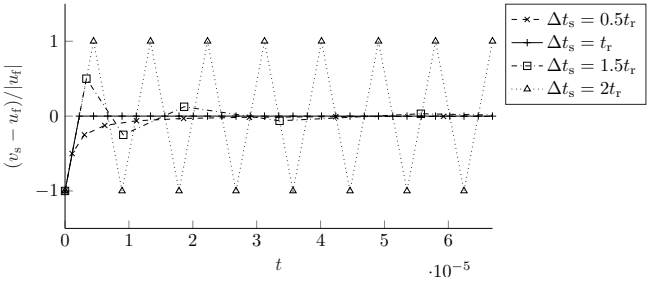


Figure 4.11: Result of applying a first order explicit time integration to the momentum equation of a single particle immersed in a constant velocity fluid and moving only due to the influence of drag. The chosen drag correlation is the one by Rong et al. [125] and the parameters are as follows: $\rho_\mathrm{f} = 1000\,\mathrm{kg/m^3}$, $\mu = 8.9 \times 10^{-4}\,\mathrm{Pa\,s}$, $D_\mathrm{s} = 1 \times 10^{-3}\,\mathrm{m}$, $\rho_\mathrm{s} = 2500\,\mathrm{kg/m^3}$, $\varepsilon_\mathrm{f} = 1$. The time step size has been recalculated after each time step as fraction or multiple of the particle relaxation time (see legend).

Here we denoted by $\boldsymbol{u}_{\mathrm{f},i}$ the fluid velocity approximation at the particle position, which we assume to be constant with respect to time in this analysis. If we further assume that $\alpha_i > 0$ is constant, the above equation has the analytical solution

$$\boldsymbol{v}_{\mathrm{s},i}(t) = \boldsymbol{u}_{\mathrm{f},i} + (\boldsymbol{v}_{\mathrm{s},i}(t=0) - \boldsymbol{u}_{\mathrm{f},i}) e^{-\frac{t}{t_{\mathrm{r},i}}} \tag{4.6.2}$$

where we have denoted by t_r the *particle relaxation time* which is defined as

$$t_{\mathrm{r},i} = \frac{m_{\mathrm{s},i}}{\alpha_i} \tag{4.6.3}$$

Clearly, t_r represents a characteristic time of the exponential decay of velocity difference between particle and fluid, i.e. it describes the characteristic time frame within which an initially resting particle is accelerated to the fluid velocity due to drag. We note that the relaxation time in terms of the drag expressions presented in Section 3.4 is given by (cf. [10, 168])

$$t_{\mathrm{r},i} = \frac{\rho_\mathrm{s} \varepsilon_{\mathrm{s},i}}{\beta_i} \tag{4.6.4}$$

or

$$t_{\mathrm{r},i} = \frac{4}{3} \frac{\rho_\mathrm{s}}{\rho_\mathrm{f}} \frac{D_{\mathrm{s},i}}{C_{\mathrm{d},i}} \frac{1}{\|\boldsymbol{u}_{\mathrm{f},i} - \boldsymbol{v}_{\mathrm{s},i}\| \varepsilon_{\mathrm{f},i}^{2-\chi_i}} \tag{4.6.5}$$

Using an explicit time integration scheme, this clearly leads to a restriction of the form

$$\Delta t_\mathrm{s} \leq C_\mathrm{r} t_{\mathrm{r},i}, \qquad \forall i \in \{1, \ldots, N_\mathrm{s}\} \tag{4.6.6}$$

with a constant $C_\mathrm{r} \in \mathbb{R}_{>0}$ that is chosen in order to ensure stability. For schemes which are of first order in the velocity, such as the explicit or semi-implicit Euler scheme and the Taylor scheme in Section 4.1.5, stability is ensured for $C_\mathrm{r} \leq 2$. In the case of a first order forward difference approximation and under the assumption of constant α we obtain

$$\boldsymbol{v}_{\mathrm{s},i}^{n+1} = \boldsymbol{v}_{\mathrm{s},i}^{n} + \frac{\Delta t_\mathrm{s}}{t_{\mathrm{r},i}^{n}} \left(\boldsymbol{u}_{\mathrm{f},i} - \boldsymbol{v}_{\mathrm{s},i}^{n} \right) \tag{4.6.7}$$

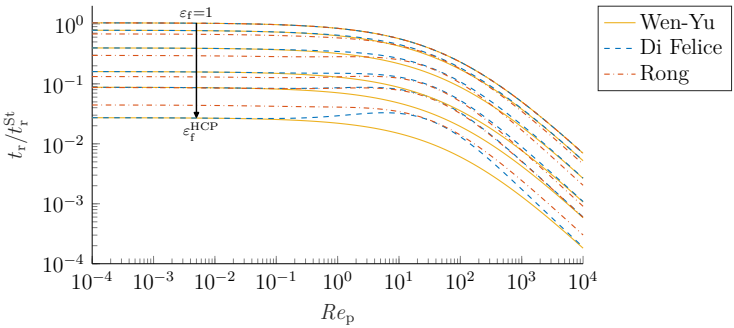


Figure 4.12: Plot of relaxation time relative to the case of Stokes drag. Plotted volume fractions for Di Felice and Rong drag from bottom to top: $\varepsilon_\mathrm{f} \in \{\varepsilon_\mathrm{f}^\mathrm{HCP}, 0.4, 0.5, 0.7, 0.9, 1.0\}$. The value $\varepsilon_\mathrm{f}^\mathrm{HCP} \approx 0.2595$ represents the limit of densest packing (see Appendix B). Note that curves are overlapping for $\varepsilon_\mathrm{f} = 1$.

From this we directly see that $\boldsymbol{v}_{\mathrm{s},i}^{n+1} = \boldsymbol{u}_{\mathrm{f},i}$ in the case of $\Delta t_\mathrm{s} = t_{\mathrm{r},i}^n$. For other choices of the time step size we have plotted the result of successively applying the forward difference in Figure 4.11. We observe that the velocity overshoots $\boldsymbol{u}_{\mathrm{f},i}$ for $t_{\mathrm{r},i}^n < \Delta t_\mathrm{s}$ while it correctly reflects the exponential decay for $\Delta t_\mathrm{s} < t_{\mathrm{r},i}^n$. Thus, while the scheme does not diverge for $C_\mathrm{r} = 2$, the velocity oscillates around the fluid velocity instead of following the one-sided exponential decay of the exact solution as long as $C_\mathrm{r} > 1$. This behavior is clearly undesirable and we therefore ensure (4.6.6) with a conservative choice of $\Delta t_\mathrm{s} \leq 0.5 t_\mathrm{r}$ in our numerical simulations. We note that other authors also propose fractions of t_r which are far smaller, such as the choice of $\Delta t_\mathrm{s} = t_\mathrm{r}/20$ in [124]. However, to the best of our knowledge, the only necessary bound is $C_\mathrm{r} < 1$, which is also implied by the lack of critical discussion of specific values of C_r in CFD-DEM literature that presents this time step restriction [10, 124, 168].

To provide some intuition on the relaxation times which have to be expected in different coupled simulations, let us analyze how the relaxation time changes with respect to porosity and particle Reynolds number. First, we consider the Stokes drag on an isolated sphere, i.e. without voidage function term $\varepsilon_{\mathrm{f},i}^{2-\chi_i}$ and with

$$C_{\mathrm{d},i} = \frac{24}{Re_{\mathrm{p,i}}} = \frac{24\mu}{\rho_\mathrm{f} \|\boldsymbol{u}_{\mathrm{f},i} - \boldsymbol{v}_{\mathrm{s},i}\| D_{\mathrm{s},i}} \tag{4.6.8}$$

In this case the relaxation time clearly simplifies to (cf. [10, 121, 124])

$$t_{\mathrm{r},i}^\mathrm{St} = \frac{\rho_\mathrm{s} D_{\mathrm{s},i}^2}{18\mu} \tag{4.6.9}$$

which we observe is independent of both, the volume fraction and particle Reynolds number. We further realize that the more general relaxation time (4.6.5) is easily expressed as

$$\frac{t_\mathrm{r}}{t_\mathrm{r}^\mathrm{St}} = \frac{24}{C_\mathrm{d}\, Re_\mathrm{p}\, \varepsilon_\mathrm{f}^{1-\chi}} \tag{4.6.10}$$

which, in contrast to the above, only depends on volume fraction and Reynolds number. In Figure 4.12 we have plotted the correction factor to the Stokes drag relaxation time (4.6.10) for a range of different particle Reynolds numbers and porosities. The chosen drag correlations coincide with those that are found in Table 3.1. Clearly, this is tightly connected

to our analysis in Appendix C. The plot shows that for small particle Reynolds numbers, $Re_\mathrm{p} < 1$, the relaxation time is only weakly dependent on the Reynolds number. In this regime, the correlations can lead to a relaxation time as small as $t_\mathrm{r} \approx t_\mathrm{r}^\mathrm{St}/50$ for Wen-Yu and Di Felice drag or $t_\mathrm{r} \approx t_\mathrm{r}^\mathrm{St}/25$ for Rong drag, depending on the considered volume fraction. For larger Reynolds numbers, the ratio further decreases and thus the time step restriction becomes more stringent. How this criterion compares to the one obtained from contact models, will be analyzed at a later point when we discuss our numerical results and may thus provide meaningful parameter choices. Before we move on to the influence of drag on the fluid phase time step, we want to note that the true relaxation time of a particle is actually larger. This is due to the fact that α_i, which we assumed to be constant, decreases together with the velocity difference between fluid and solid. Thus, while it provides a useful metric for determining suitable time steps, it does not exactly correspond to the physically observable time scale.

4.6.2 Fluid phase time step restriction

In Section 4.3 we motivated an implicit treatment of drag within the fluid phase due to the time step restriction that emerges from an explicit term. Here we want to provide more detail on this aspect. To this end we consider (4.3.10) and replace α by reordering (4.6.3). Thus, with the same assumptions as in the section on implicit drag calculation, one can approximate

$$\frac{\boldsymbol{f}_\mathrm{d}}{\varepsilon_\mathrm{f}\rho_\mathrm{f}} \approx \frac{n_\mathrm{p}\langle\frac{m_\mathrm{s}}{t_\mathrm{r}}\rangle^\mathrm{p}}{\rho_\mathrm{f}\varepsilon_\mathrm{f}}(\boldsymbol{u}_\mathrm{f} - \boldsymbol{u}_B) \tag{4.6.11}$$

If we further assume that the particles have identical volume and density, so that $\varepsilon_\mathrm{s} = n_\mathrm{p}V_\mathrm{s}$, we obtain

$$\frac{\boldsymbol{f}_\mathrm{d}}{\varepsilon_\mathrm{f}\rho_\mathrm{f}} \approx \frac{\varepsilon_\mathrm{s}}{\varepsilon_\mathrm{f}}\frac{\rho_\mathrm{s}}{\rho_\mathrm{f}}\langle\frac{1}{t_\mathrm{r}}\rangle^\mathrm{p}(\boldsymbol{u}_\mathrm{f} - \boldsymbol{u}_B) \tag{4.6.12}$$

Thus, if we consider the fluid momentum equation in Lagrangian form and with only the drag term on the right-hand side, we obtain a fluid relaxation time

$$t_\mathrm{rf} = \frac{\varepsilon_\mathrm{f}}{\varepsilon_\mathrm{s}}\frac{\rho_\mathrm{f}}{\rho_\mathrm{s}}\left(\langle\frac{1}{t_\mathrm{r}}\rangle^\mathrm{p}\right)^{-1} \tag{4.6.13}$$

By replacing the particle phase average with the reciprocal of the minimum relaxation time within the solid phase, we can infer the rather conservative time step restriction of the form

$$\Delta t_\mathrm{f} \leq C_\mathrm{r}\frac{\varepsilon_\mathrm{f}}{\varepsilon_\mathrm{s}}\frac{\rho_\mathrm{f}}{\rho_\mathrm{s}}t_\mathrm{r}^\mathrm{min} \tag{4.6.14}$$

where

$$t_\mathrm{r}^\mathrm{min} = \min_{j\in\{1,\dots,N_\mathrm{s}\}} t_{\mathrm{r},j} \tag{4.6.15}$$

We note that in the case of Stokes drag and monodisperse particles, the relaxation time is identical for all particles and we obtain the time step restriction (cf. [121])

$$\Delta t_\mathrm{f} \leq C_\mathrm{r}\frac{\varepsilon_\mathrm{f}}{\varepsilon_\mathrm{s}}\frac{\rho_\mathrm{f}D_{\mathrm{s},i}^2}{18\mu} \tag{4.6.16}$$

Clearly, the factor $\varepsilon_\mathrm{f}/\varepsilon_\mathrm{s}$ only tightens the time step restriction if $\varepsilon_\mathrm{f} < 0.5$. The smallest value of this term, obtained for the volume fraction in a HCP, is approximately 0.35 i.e. the factor causes a maximum reduction of time step to about one third of the particle relaxation time.

For values $\varepsilon_\mathrm{f} > 0.5$ the time step restriction is relaxed and eventually vanishes since the factor is unbounded for $\varepsilon_\mathrm{f} \to 1$. The factor $\rho_\mathrm{f}/\rho_\mathrm{s}$ in (4.6.14), on the other hand, typically reduces the allowed time step in comparison to the solid phase restriction and can have significant influence in cases of large density difference, such as in gas-solid flows. As with the time step restriction in the solid phase, we will compare this to the previously considered CFL condition when we present our numerical results.

At this point we want to conclude our discussion on time step restrictions and re-iterate that the above criterion (4.6.14) is only of importance for an explicit treatment of drag in the fluid phase. In Table 4.1 we have collected all the time step restriction that have been discussed in earlier chapters and in this section to provide a convenient overview.

Phase	**Origin**	**Restriction**	**Notes**
Solid phase	Contact model (LSD)	$\Delta t_\mathrm{s} \leq C t_\mathrm{c}$, $C \leq 0.2$	Critical time t_c from (4.1.77) Choices of C discussed in Section 4.1.5
	Contact model (Hertz)	$\Delta t_\mathrm{s} \leq \frac{t_\mathrm{d}}{N_\mathrm{t,d}}$, $N_\mathrm{t,d} \geq 15$	Contact duration t_d from (4.1.83) Lower $N_\mathrm{t,d}$ bound from $C = 0.2$ in LSD
	Drag force	$\Delta t_\mathrm{s} \leq C_\mathrm{r} t_\mathrm{r}$	Relaxation time from (4.6.4) or (4.6.5) Choice of $C_\mathrm{r} = 0.5$ in this work
Fluid phase	CFL	$\Delta t_\mathrm{f} \leq \mathcal{C} \frac{h}{\|\boldsymbol{u}_\mathrm{f}\|}$, $\mathcal{C} \in (0,1)$	Exact form in (2.3.3)
	Drag force	$\Delta t_\mathrm{f} \leq C_\mathrm{r} \frac{\varepsilon_\mathrm{f}}{\varepsilon_\mathrm{s}} \frac{\rho_\mathrm{f}}{\rho_\mathrm{s}} t_\mathrm{r}^\mathrm{min}$	$t_\mathrm{r}^\mathrm{min}$ from (4.6.15) Only for explicit treatment of drag

Table 4.1: Time step restrictions in both phases of a coupled CFD-DEM simulation.

4.7 Substepping procedure and full algorithm

As we will see in the upcoming chapter and already hinted at in earlier discussions, the time step size allowed within a soft-sphere DEM simulation is typically significantly smaller than any restriction on the fluid phase time step. However, since the DEM time step is chosen in such a way as to resolve each collision in numerous steps, the resulting variations in particle position between two DEM time steps are small on the continuous phase scale. It is therefore desirable to recalculate the fluid motion only after changes have occurred that are relevant for the coupling. We achieve this by employing the common approach of DEM substepping (cf. [21, 168, 172]). This means, that within the time interval $[t^n, t^{n+1}]$, where $\Delta t_\mathrm{f} = t^{n+1} - t^n$ is the time step for the continuous phase, we execute all steps of the solid phase solver N_sub times. Clearly, each time integration within this substepping is done with a time step

$$\Delta t_\mathrm{s} = \frac{\Delta t_\mathrm{f}}{N_\mathrm{sub}} \tag{4.7.1}$$

Since we do not recompute fluid phase quantities after a DEM step is done, the coupling force $\boldsymbol{F}_{\mathrm{fs},i}^{n_k}$ at time level $t^{n_k} = t^n + k\Delta t_\mathrm{s}$, $k = 1, \ldots, N_\mathrm{sub}$, is calculated based on fluid quantities at time level t^n. However, since the particles move, both, the result of interpolation from the fluid phase and more importantly the solid particle velocity, still change. Consequently, we recompute the coupling forces acting onto a particle in every DEM step. For the sake of completeness, we note that one could update the drag in yet another time interval. This time interval is clearly determined by the above considerations on time step restrictions

due to relaxation times associated with drag. Thus, it is only relevant in the case where this time step restriction is in between the DEM time step and Δt_f. Within the numerical simulations presented in the upcoming chapter, we have not pursued this approach, we do however acknowledge that this aspect certainly bears some potential for future extensions of our coupled algorithm.

So let us assume that we only consider a single substepping loop with time step size chosen according to the DEM time step restriction. Without modification and only taking into account the coupling forces at the end of DEM substepping, this procedure would violate Newton's third law since the change in momentum $\Delta t_\mathrm{f} \boldsymbol{f}_\mathrm{fs}$ is not equal the overall momentum change per unit volume of particles during the complete time step Δt_f. The total change of momentum due to coupling forces for a particle i is given by

$$\sum_{k=1}^{N_\mathrm{sub}} \boldsymbol{F}_{\mathrm{fs},i}^{n_k} \Delta t_\mathrm{s} = \left(\frac{1}{N_\mathrm{sub}} \sum_{k=1}^{N_\mathrm{sub}} \boldsymbol{F}_{\mathrm{fs},i}^{n_k} \right) \Delta t_\mathrm{f} \tag{4.7.2}$$

which means that, compared to the calculation of $\boldsymbol{f}_\mathrm{fs}$ without substepping in the DEM phase, one only has to replace the force acting on the particles during $[t^n, t^{n+1}]$ by the average of values obtained from substeps. This suffices to restore the conservation property in the explicit treatment of drag. In case of an implicit treatment, we ensure the same by using

$$\alpha = \frac{1}{N_\mathrm{sub}} \sum_{k=1}^{N_\mathrm{sub}} \alpha_i^{n_k} \tag{4.7.3}$$

$$\alpha \boldsymbol{v}_\mathrm{s} = \frac{1}{N_\mathrm{sub}} \sum_{k=1}^{N_\mathrm{sub}} \alpha_i^{n_k} \boldsymbol{v}_{\mathrm{s},i}^{n_k} \tag{4.7.4}$$

in (4.3.10) for the drag components and the above average for all remaining forces.

With the substepping being explained, we have now discussed every component of the complete coupled scheme. To wrap this chapter up, we provide an algorithmic overview of the implemented algorithm in Figure 4.13. This chart is intended to clarify the workflow in more detail than the rough outline at the beginning of this chapter. In particular, it also visualizes the substepping procedure discussed within the current section and shows an aspect that we did neglect so far. Namely the fact that an additional coarse graining step is necessary before the very first time step in order to provide initial values for the volume fraction.

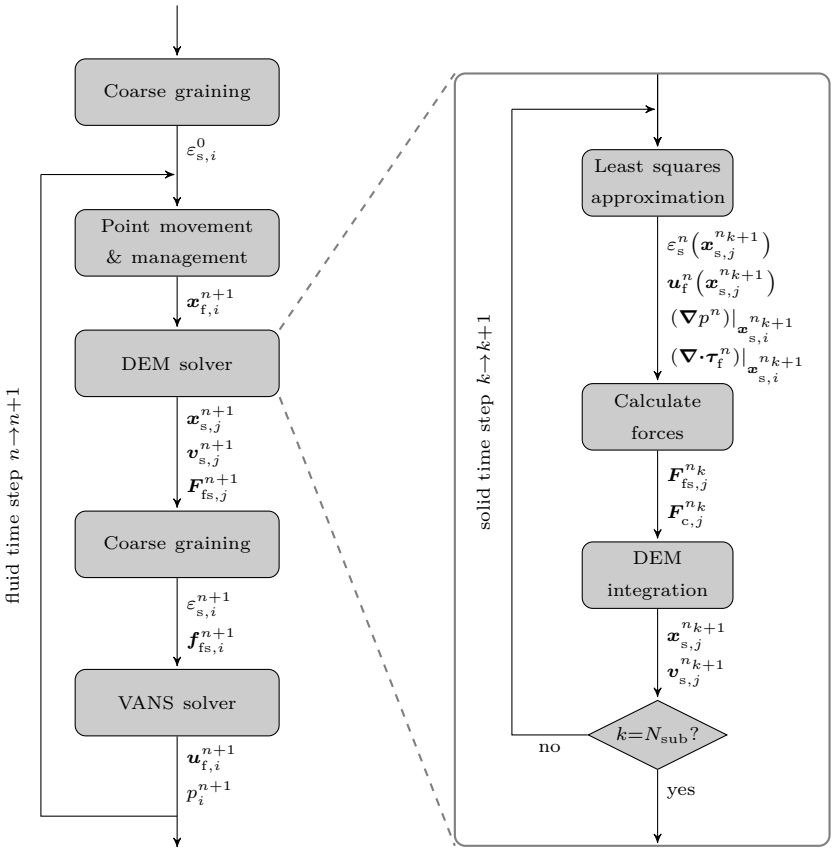


Figure 4.13: Algorithmic overview of our CFD-DEM algorithm. On the left, all major steps within a fluid time step are shown. The "DEM solver" step is visualized in more detail on the right, in order to clarify the substepping procedure. The outputs of each step within the algorithm are displayed next to the subsequent arrow. Quantities attached to fluid points are denoted by a subscript $i \in \{1, \ldots, N_{\mathrm{f}}\}$ while those attached to solid particles are denoted by the subscript $j \in \{1, \ldots, N_{\mathrm{s}}\}$.

Chapter 5

Numerical results

With the previous chapter we have fully clarified all components of our extension to the FPM. Clearly, we still owe the reader numerical proof of the fact that the presented scheme is able to produce satisfactory results in its intended field of application, i.e. the simulation of two-way coupled fluid-solid flows with particulate solid phase. Therefore the final chapter of this work will be devoted to this task. Due to the fact that we did not tailor our approach towards one specific application, we are faced with an overwhelming amount of literature on fluid-solid flow phenomena, which can provide us with experimental or numerical results that act as benchmarks for our scheme. For validation purposes, we did however conclude that trying to reproduce an experimental result which has not yet been numerically reproduced by a similar coupling in a different numerical framework bears the risk of exceeding the limits of either the unresolved coupling in general, or closures for drag and unresolved stress terms in particular. Based on this notion we have chosen three simulation setups where it has already been shown that satisfactory results can be obtained through numerical simulation. In the following these choices are considered in order of increasing complexity.

5.1 Single particle sedimentation

We start with the test case of a single (spherical) particle that is sedimenting under the effect of gravity in a container filled with fluid, which is a commonly used example in the context of CFD-DEM (cf. [52, 100, 124]). We assume that the particle Reynolds number

$$Re_\mathrm{p} = \frac{\varepsilon_\mathrm{f}\rho_\mathrm{f}D_\mathrm{s}\|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\|}{\mu} \tag{5.1.1}$$

is small enough for the drag to be described by Stokes law

$$\boldsymbol{F}_\mathrm{d} = 3\pi\mu D_\mathrm{s}\boldsymbol{u}_\mathrm{sf} \tag{5.1.2}$$

so that an analytical solution can be obtained for the motion in a quiescent fluid. We can compare this exact result to numerical results from one-way and two-way coupled simulations based on the numerical scheme proposed in Chapter 4. The purpose of this validation test case is to test that drag forces are calculated and integrated in the DEM code correctly and that the scheme does not produce distortions of the fluid flow which are unacceptable given the small particle size.

To calculate the analytical solution, we assume that the solid volume fraction is negligible and the fluid remains at rest, so that

$$\boldsymbol{u}_\mathrm{sf} = \varepsilon_\mathrm{f}(\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}) = -\boldsymbol{v}_\mathrm{s} \tag{5.1.3}$$

If no collisions with the container walls occur within the simulated time frame, Newton's equation for a single particle under the effect of constant gravity reads

$$\frac{\mathrm{d}\boldsymbol{v}_\mathrm{s}}{\mathrm{d}t} = \frac{1}{m_\mathrm{s}}\boldsymbol{F}_\mathrm{fs} + \boldsymbol{g} \tag{5.1.4}$$

For the quiescent fluid the fluid-solid interaction force consist only of the pressure gradient contribution and Stokes drag, i.e.

$$\boldsymbol{F}_{\mathrm{fs}} = \boldsymbol{F}_{\mathrm{p}} + \boldsymbol{F}_{\mathrm{d}} = -\boldsymbol{\nabla} p V_{\mathrm{s}} - 3\pi\mu D_{\mathrm{s}} \boldsymbol{v}_{\mathrm{s}} \tag{5.1.5}$$

The pressure gradient in a non-moving fluid under the effect of gravity reduces to

$$\boldsymbol{\nabla} p = \rho_{\mathrm{f}} \boldsymbol{g} \tag{5.1.6}$$

which clearly contributes to buoyancy in the solid phase. If we plug these results into (5.1.4) and consider only the velocity component along the direction of gravity, we obtain

$$\frac{\mathrm{d}v_{\mathrm{s}}}{\mathrm{d}t} = -\frac{3\pi\mu D_{\mathrm{s}}}{\rho_{\mathrm{s}} V_{\mathrm{s}}} v_{\mathrm{s}} + \frac{(\rho_{\mathrm{s}} - \rho_{\mathrm{f}})g}{\rho_{\mathrm{s}}} \tag{5.1.7}$$

The solution to this ordinary differential equation with initial condition $v_{\mathrm{s}}(0) = 0$ is of the form

$$v_{\mathrm{s}}(t) = \left(1 - e^{-\frac{t}{t_{\mathrm{r}}}}\right) v_{\mathrm{T}} \tag{5.1.8}$$

where t_{r} actually represents the Stokes relaxation time

$$t_{\mathrm{r}}^{\mathrm{St}} = \frac{\rho_{\mathrm{s}} D_{\mathrm{s}}^2}{18\mu} \tag{5.1.9}$$

as in Section 4.6 and v_{T} denotes the terminal velocity under Stokes drag

$$v_{\mathrm{T}} = \frac{(\rho_{\mathrm{s}} - \rho_{\mathrm{f}}) D_{\mathrm{s}}^2 g}{18\mu} \tag{5.1.10}$$

Clearly, this means that the particle does not accelerate beyond v_{T}, which marks the point where drag and gravitational force balance each other out. Additionally, the velocity difference $|v_{\mathrm{s}} - v_{\mathrm{T}}|$ follows an exponential decay with a characteristic time scale defined by the relaxation time $t_{\mathrm{r}}^{\mathrm{St}}$, which we use to define a dimensionless time scale on which we analyze our results in the following. Now that we have derived the reference solution which we use for quantitative comparisons, we want to specify the setup of our numerical simulations.

Simulation setup

The flow geometry and particle diameter are identical to the ones used in the paper by Robinson et al. [124] and are visualized in Figure 5.1. The width and height are chosen so that walls are not expected to influence the flow around the particle. The particle is initially centered within the column on the x-y-plane and placed at a height of $z_0 = 40 D_{\mathrm{s}}$ (along the z-axis). The fluid under consideration is water, while the particle density is 2.5 times that of water which does, for example, match the density of glass beads. The exact parameters can be found in Table 5.1. If we use these parameters to determine the terminal velocity in (5.1.10) and plug this together with $\varepsilon_{\mathrm{f}} \leq 1$ into (5.1.1), we obtain $Re_{\mathrm{p}} \leq 1$, which means that Stokes drag can be considered valid.

The velocity boundary condition for the fluid phase is set to no-slip and is independent of the coupling, while the pressure requires a more careful consideration. Following the derivations of Section 4.4, the Poisson equation for hydrostatic pressure in a single-phase setting reads

$$\boldsymbol{\nabla} \cdot \left(\frac{1}{\rho_{\mathrm{f}}} \boldsymbol{\nabla} p_{\mathrm{hyd}} \right) = \boldsymbol{\nabla} \cdot \boldsymbol{g} \tag{5.1.11}$$

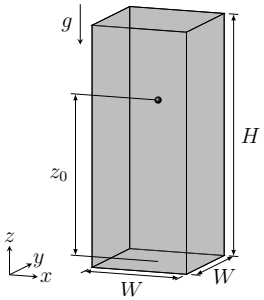


Figure 5.1: Geometry of single particle sedimentation test case (not to scale).

	Symbol	Unit	Value
Cylinder width	W	m	4×10^{-3}
Cylinder height	H	m	6×10^{-3}
Initial position	z_0	m	4×10^{-3}
Particle diameter	D_s	m	1×10^{-4}
Solid density	ρ_s	kg/m^3	2500
Fluid density	ρ_f	kg/m^3	1000
Dynamic viscosity	μ	Pa s	8.9×10^{-4}
Gravity	g	m/s^2	9.81

Table 5.1: Parameters of single particle sedimentation test case.

In the FPM this equation is typically supplemented with the boundary condition

$$\frac{\partial p_\mathrm{hyd}}{\partial n} \coloneqq \boldsymbol{n} \cdot \boldsymbol{\nabla} p_\mathrm{hyd} = \rho_\mathrm{f}(\boldsymbol{n} \cdot \boldsymbol{g}) \tag{5.1.12}$$

at walls and inflow boundaries, with $\boldsymbol{n}$ denoting the inward pointing boundary normal. In our coupled scheme, we recall from (4.4.11) that the density is replaced by the generalized notation $\tilde{\rho_\mathrm{f}}$ from (4.4.8) and either of two changes is introduced by the coupling:

1. An additional term $\boldsymbol{\nabla} \cdot (B\boldsymbol{u}_\mathrm{B})$ is added on the right-hand side of (5.1.11)
2. The gravity term in (5.1.11) is replaced by $\tilde{\boldsymbol{g}} = \boldsymbol{g} + \boldsymbol{f}_\mathrm{fs}$

In analogy to (5.1.12) we can account for these additional terms by enforcing

$$\frac{\partial p_\mathrm{hyd}}{\partial n} = \tilde{\rho_\mathrm{f}}(\boldsymbol{n} \cdot \tilde{\boldsymbol{g}}) + \tilde{\rho_\mathrm{f}} B(\boldsymbol{n} \cdot \boldsymbol{u}_\mathrm{B}) \tag{5.1.13}$$

In the current setup, the averaging kernel is, however, cut off at a distance smaller than that between the particle and closest wall. Therefore, we obtain $\tilde{\rho_\mathrm{f}} = \rho_\mathrm{f}$, $\boldsymbol{f}_\mathrm{fs} = 0$, $B = 0$ close to all walls at any time during the simulation, which ultimately restores the single-phase boundary condition. The same can be observed for the dynamic pressure. To derive the corresponding boundary condition, we take the dot product of

$$\frac{\mathrm{D}\boldsymbol{u}_\mathrm{f}}{\mathrm{D}t} = -\frac{1}{\tilde{\rho_\mathrm{f}}}(\boldsymbol{\nabla} p_\mathrm{hyd} + \boldsymbol{\nabla} p_\mathrm{dyn}) + \frac{1}{\tilde{\rho_\mathrm{f}}} \boldsymbol{\nabla} \cdot \boldsymbol{\tau}_\mathrm{f}(\boldsymbol{u}_\mathrm{f}) - B(\boldsymbol{u}_\mathrm{f} - \boldsymbol{u}_\mathrm{B}) + \tilde{\boldsymbol{g}} \tag{5.1.14}$$

with the boundary normal and use the fact that the hydrostatic pressure term cancels out with those terms already accounted for by (5.1.13). With the above observation of vanishing coupling terms close to the walls, we are once more left with the single-phase boundary condition

$$\frac{\partial p_\mathrm{dyn}}{\partial n} = -\rho_\mathrm{f}\left(\boldsymbol{n} \cdot \frac{\mathrm{D}\boldsymbol{u}_\mathrm{f}}{\mathrm{D}t}\right) + \boldsymbol{n} \cdot (\boldsymbol{\nabla} \cdot \boldsymbol{\tau}_\mathrm{f}(\boldsymbol{u}_\mathrm{f})) \tag{5.1.15}$$

Since we apply no-slip conditions to all walls, the first term on the right-hand side of (5.1.15) always vanishes. As for the second term, this can be calculated as usual, i.e. based on the finite difference stencil and velocities at points close to the boundary.

Within the considered time frames, the particle does not collide with the bottom wall, so that the contact model is not relevant for this test case and in particular it does not impose

its strong time restriction. However, as we have seen in Section 4.6, it is necessary to choose the solid phase time step as a fraction of the relaxation time. With the parameters from Table 5.1 we obtain

$$t_{\mathrm{r}}^{\mathrm{St}} = \frac{\rho_{\mathrm{s}} D_{\mathrm{s}}^2}{18\mu} \approx 1.56 \times 10^{-3}\,\mathrm{s} \tag{5.1.16}$$

which forms an upper bound for our solid phase time step. The specific choice of time step size in our simulations is discussed in the following. We note that for all of the simulations in this chapter, the time integration in the solid phase follows the semi-implicit Euler scheme discussed in Section 4.1.5.

Results of one-way coupling

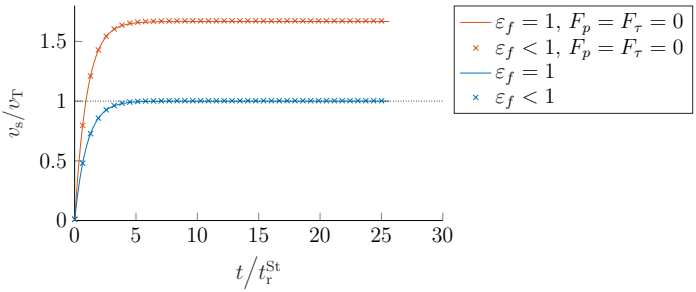


Figure 5.2: Sedimentation velocity of the solid particle from simulations with one-way coupling. The case of evaluating the drag force with $\varepsilon_{\mathrm{f}} = 1$ is depicted by a line while the case of using the value $\varepsilon_{\mathrm{f}} < 1$ calculated by the coarse graining algorithm is depicted by markers. The color distinguishes whether the pressure gradient and viscous stress force have been included or not.

First, let us consider the case of one-way coupling, i.e. with no influence of the solid phase on the fluid phase. In this case the only sources of errors compared to our analytical solution are due to non-zero fluid velocity and the fact that the fluid volume fraction is smaller than one. In Figure 5.2 we have plotted the relative solid particle velocity over the simulated time frame $t \in [0, 0.04\,\mathrm{s}]$. The simulation was performed with $h = 3D_{\mathrm{s}}$ and a rather conservative timestep $\Delta t_{\mathrm{f}} = \Delta t_{\mathrm{s}} = 1 \times 10^{-5}\,\mathrm{s} \approx t_{\mathrm{r}}^{\mathrm{St}}/156$ in both phases. We elected to consider both, the results obtained by keeping $\varepsilon_{\mathrm{f}} = 1$ and by calculating the solid volume fraction with a Gaussian kernel of bandwidth $b = 3D_{\mathrm{s}}$ which is cut off at a distance $c = 3b$. The results show that, as long as the non-drag forces are included, the expected exponential increase of velocity towards the theoretical value of the sedimentation velocity is correctly obtained in both the case with a fixed value of $\varepsilon_{\mathrm{f}} = 1$ and in the case where the volume fraction is calculated by the coarse graining. If the other coupling forces are neglected, the sedimentation velocity exceeds the analytically determined value. As is to be expected in the one-way coupled case, the simulations have shown that the viscous stress force $\boldsymbol{F}_\tau$ has negligible magnitude in comparison to both, the drag and pressure gradient force. Neglecting the pressure gradient force corresponds to omitting the buoyancy contribution, which in turn explains the over-predicted solid particle velocity.

To see which influence the spatial resolution had on the above results, we calculated the error in solid particle velocity for a range of smoothing lengths. This is shown in Figure 5.3 where we have considered $\Delta t_{\mathrm{f}} = \Delta t_{\mathrm{s}} = 1 \times 10^{-4}\,\mathrm{s}$ together with $h = 2^i D_{\mathrm{s}}$, $i \in \{0, 1, 2, 3\}$. These choices of smoothing length lead to the following numbers of points in the fluid phase:

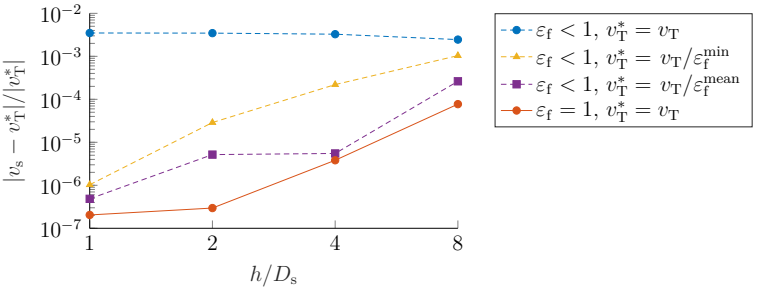


Figure 5.3: Error in sedimentation velocity obtained from one-way coupled simulations and averaged over the interval $t \in [20t_r^{St}, 25t_r^{St}]$. The simulations were repeated with a fixed time step size ($\Delta t = 1 \times 10^{-4}$ s) and varying smoothing length. The line style distinguishes the case of evaluating the drag force with $\varepsilon_f = 1$ or with a value $\varepsilon_f < 1$ calculated by the coarse graining algorithm. The marker style distinguishes whether the particle velocity is compared to v_T from (5.1.10) or v_T/ε_f.

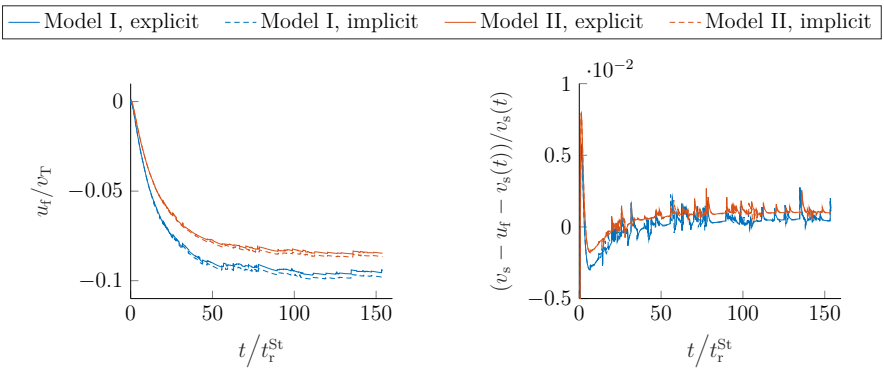
h/D_s	1	2	4	8
N_f	1 764 815	256 646	40 551	6947

Table 5.2: Number of fluid points for different smoothing lengths in the one-way coupled FPM simulations of the single particle sedimentation test case. The point count varies during runtime due to point addition and deletion.

We note that the time step has been chosen larger than in the simulations for Figure 5.2 since we observed that the error was insensitive to this choice. We performed simulations with drag force calculated exactly as specified in (5.1.5) as well as based on the superficial velocity, i.e. using the non-zero volume fraction obtained from coarse graining. In Figure 5.3 the corresponding results are distinguished by line style. Since strictly speaking the analytical solution in the case of $\varepsilon_f < 1$ is given by v_T/ε_f, we compare the corresponding results to v_T, v_T/ε_f^{min} and v_T/ε_f^{mean}. Here we denote by ε_f^{min} the value obtained by evaluating the averaging kernel at the origin, i.e.

$$\varepsilon_f^{min} = \tilde{w}(0)V_s = \frac{V_s}{(\pi b^2)^{\frac{3}{2}}} \tag{5.1.17}$$

and ε_f^{mean} denotes the mean volume fraction at the particle position throughout the simulation. Figure 5.2 illustrates that using a non-zero solid volume fraction during drag calculation introduces an error when comparing to the sedimentation velocity from (5.1.10). If instead we consider the error relative to $v_T^* = v_T/\varepsilon_f$ with any of the above definitions of volume fraction, we obtain significantly lower errors. It is clearly shown that on the one hand, the error decreases together with the smoothing length and on the other hand it is dwarfed by the error due to non-zero solid volume fraction, if it is determined with respect to v_T. In conclusion we observed that even the errors obtained from a direct comparison to v_T are within the same magnitude reported in [124]. There the authors did however not present the additional errors when comparing to the analytical solutions that take volume fraction into account.



(a) Fluid velocity at particle centroid position relative to Stokes drag terminal velocity.

(b) Relative error in realtive sedimentation velocity $v_s - u_f$ from simulations compared to $v_s(t)$ from (5.1.8).

Figure 5.4: Transient results from simulations with two-way coupling and $h = 3D_s$, $\Delta t_f = \Delta t_s = 1 \times 10^{-4}$ s. The choice of VANS model, as discussed in Section 3.3, is distinguished by color while the treatment of drag, as discussed in Section 4.3, is distinguished by line style.

Results of two-way coupling

Next, we want to discuss the results obtained from two-way coupled simulations. Similar to the above investigation, let us first consider the transient behavior. Again, we choose both, the smoothing length and averaging bandwidth, to be three times the particle diameter and use an identical time step $\Delta t = 1 \times 10^{-4}\,\mathrm{s} \approx t_r^{St}/16$ in both phases. In Figure 5.4 we have plotted the results of these simulations for different combinations of the coupling model discussed in Section 3.3 and both explicit and implicit drag calculation in the fluid phase (cf. Section 4.3). From Figure 5.4a we observe that the fluid points around the particle location are accelerated to about $8 - 10\%$ of the Stokes terminal velocity from (5.1.10). Thus, since the fluid is no longer approximately at rest, we compared the relative velocity $v_s - u_f$ to our analytical result in Figure 5.4b. There we observe that, aside from some initial fluctuations, the error relative to the analytical solution is well below 1% throughout the simulated time frame. Furthermore, as should be the case, there is little difference between the results from an explicit and implicit treatment of drag in both plots. There is however a difference in fluid velocity between the two CFD-DEM formulations, formerly referred to as model I and II (see Section 3.3). Due to this acceleration of fluid around the particle occurring on a significantly larger time scale than the relaxation in the one-way coupled case, we analyze the average error in relative velocity over the range $t \in [100t_r^{St}, 150t_r^{St}]$, instead of $t \in [20t_r^{St}, 25t_r^{St}]$, when varying the spatial resolution.

As the careful reader may have noticed, we have compared the transient results above to the analytical solution obtained under the assumption of negligible solid volume fraction. While comparing the particle velocity to the analytical terminal velocity modified by the porosity was clearly adequate in the one-way coupled case, using the modified analytical result in comparisons for the two-way coupled case has led to inconclusive results. A careful investigation of this is left for future work but does not detract from the quality of our results, as is shown by comparisons to [124] in the following.

As for the dependency on spatial resolution shown in Figure 5.5, we see that for relatively large smoothing lengths model II appears to be superior in terms of the error in relative

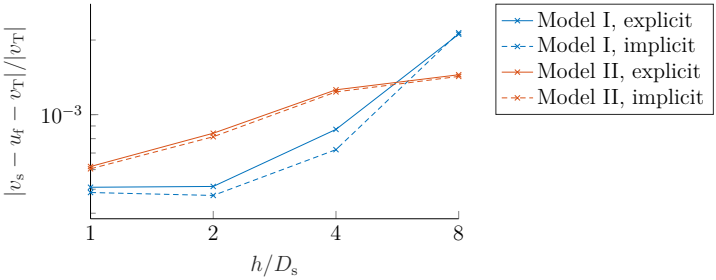


Figure 5.5: Error in sedimentation velocity obtained from two-way coupled simulations and averaged over the interval $t \in [100t_r^{St}, 150t_r^{St}]$. The simulations were repeated with a fixed time step size ($\Delta t = 1 \times 10^{-4}$ s) and varying smoothing length. The choice of VANS model, as discussed in Section 3.3, is distinguished by color while the treatment of drag, as discussed in Section 4.3, is distinguished by line style.

sedimentation velocity while with increasing number of points, model I produces smaller errors. To qualify the first observation, it is however important to note that for the very small point numbers associated with $h = 8D_s$, the fluid velocity fluctuations were significant. Nevertheless, if we take the maximum error instead of the mean over the specified time interval, the error is still below 0.3% for all smoothing lengths. Again, comparing this to the errors reported in [124], we observe that even the maximum errors in our simulations are well below those. Another interesting aspect of the results in Figure 5.5 is that in model II there is little difference between the explicit and implicit drag treatment while it does have some influence in model I. Before wrapping up this test case, we want to shortly discuss the runtimes for those simulations which lead to Figure 5.3 and Figure 5.5.

	$h/D_s = 1$	$h/D_s = 8$
One-way, $\varepsilon_f = 1$	1269 s	8 s
One-way, $\varepsilon_f < 1$	1270 s	9 s
Two-way, Model I, explicit	1241 s	8 s
Two-way, Model I, implicit	1280 s	8 s
Two-way, Model II, explicit	1217 s	10 s
Two-way, Model II, implicit	1217 s	8 s

Table 5.3: Normalized runtimes of single particle sedimentation simulations. The absolute runtime for each simulation is divided by T/t_r^{St} with T denoting the end time of the simulation.

The runtimes shown in Table 5.3 were obtained by running each simulation with 16 MPI threads on an individual dual socket node equipped with two Intel Xeon E5-2670 processors and deactivated hyper-threading. Firstly, considering the number of points for both, the minimum and maximum smoothing length (see Table 5.2), the drastic increase in runtime resulting from changes in spatial resolution, is to be expected. As noted above, to obtain similar error magnitudes to the cited reference, large smoothing lengths and thus comparatively low run times are sufficient. Secondly, we observe that the normalized runtimes for one-way and two-way coupled simulations are almost identical. Thus, there is no unreasonable computational overhead associated with the modifications to the projection scheme in the fluid phase. We further note that a difference in computational cost between the case denoted as "One-way, $\varepsilon_f = 1$" to the case denoted as "One-way, $\varepsilon_f < 1$", would have to be fully attributed to the averaging scheme. The identical run times thus verify that, for a

single particle and the considered range of spatial resolutions in the fluid phase, there is no significant computational cost involved in the averaging scheme.

5.2 Onset of fluidization

After having concluded a test case with emphasis on the solid phase motion, we now want to consider an example which shows that the influence of coupling on the fluid phase is correctly reproduced by our scheme. For this purpose we chose to investigate the phenomenon of fluidization. Fluidization describes the effect that an assembly of solid particles within a vertical flow geometry becomes suspended in the fluid phase at sufficiently large upward flow velocities. These geometries are typically cylindrical in shape and referred to as *fluidization column*, while the assembly of particles is often referred to as *bed*. In the fluidized state the forces exerted by the fluid phase onto the particles will balance out gravitational forces and the solid phase effectively behaves like a fluid, hence the name fluidization. The minimum velocity needed to induce fluidization is called *minimal* or *minimum fluidization velocity*. Once the minimum fluidization velocity is reached, the behavior may differ depending on the type of fluid and solid. While a suspension in liquid typically leads to a uniformly expanding bed, as stated by Di Felice [35], a gas fluidized bed may exhibit a range of different phenomena, such as bubbling or slugging [55, 84]. For a visualization of these different configurations we refer the reader to the classical textbook by Kunii and Levenspiel [84] and in particular the sketch on page 2 therein. We note that much of the dynamic behavior is accompanied by the ability to maintain nearly uniform temperature within a reactor even for highly exothermic reactions and also by many other advantageous properties which lead to a range of important industrial applications [84]. For now, we do not concern ourselves with these highly dynamic and complex phenomena. Instead, we only consider the behavior under constantly increasing inflow velocity up to the point where fluidization begins. This part of the process provides us with a convenient quantitative comparison for our numerical scheme, by way of the pressure drop across the fluidization column. An example which incorporates yet another type of configuration and validates the capabilities of our scheme to represent more dynamic behavior will follow in the next section. We emphasize that both, the current example and the one in the next section, employ a two-way coupling between the phases. In the current benchmark we increase the inflow velocity over time and expect the pressure difference between in- and outflow to increase up until the minimum fluidization velocity, beyond which it remains constant. The fact that the pressure drop assumes such a constant value results from the counteracting effects of drag increase due to increasing relative velocity and drag reduction due to smaller solid volume fractions when the bed gets lifted. On the one hand we may compare the pressure drop from simulations to commonly used predictive formulas. On the other hand, in order to provide an easily reproducible and accessible benchmark, we chose a simulation setup similar to the one in [57], which is also distributed together with the code of the open source CFD-DEM solver *CFDEM*[1]. We may thus further compare our results to this algorithm which couples the two open source solvers *LIGGGHTS*[2] and *OpenFOAM*[3]. At the time our investigations were conducted, the setup could also be found within the *CFDEM GitHub* repository[4].

[1] `https://www.cfdem.com/cfdemrcoupling-open-source-cfd-dem-framework`
[2] `https://www.cfdem.com/liggghtsr-open-source-discrete-element-method-particle-simulation-code`
[3] `https://www.openfoam.com/`
[4] `https://github.com/CFDEMproject/CFDEMcoupling-PUBLIC/tree/master/tutorials`

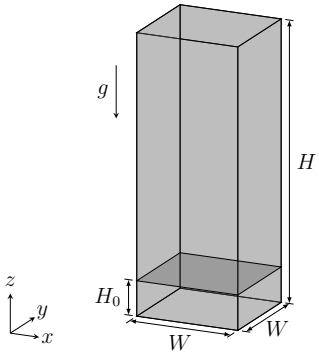


Figure 5.6: Geometry of fluidization test case (not to scale).

	Symbol	Unit	Value
Cylinder width	W	m	0.024 55
Cylinder height	H	m	0.08
Position of plate	H_0	m	0.01
Gravity	g	$\mathrm{m/s^2}$	9.81
Fluid density	ρ_f	$\mathrm{kg/m^3}$	10
Dynamic viscosity	μ	Pa s	1.5×10^{-3}
Particle number	N_s	-	1×10^4
Particle diameter	D_s	m	0.001
Solid density	ρ_s	$\mathrm{kg/m^3}$	2000
Young's modulus	Y	Pa	5×10^6
Poisson's ratio	ν	-	0.45
Coeff. of Rest.	e_n	-	0.3

Table 5.4: Parameters of fluidization test case.

Simulation setup

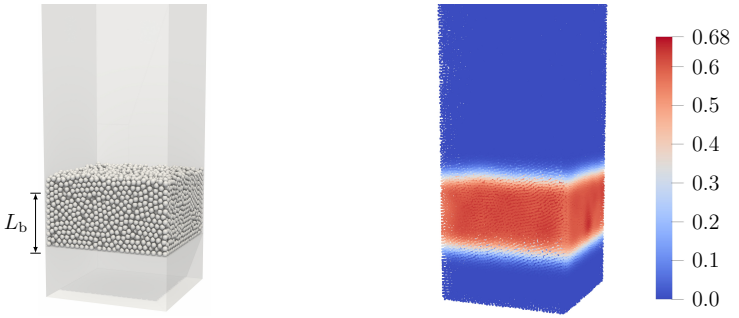
While the fluidization column considered in [57] has a circular cross section, we have chosen a cylinder with square base of identical cross-sectional area so that we can apply an averaging with the kernel mirroring described in Section 4.2. We further position a plane boundary at height H_0 from the inflow which represents a perforated wall, visible only to the solid phase and not included in the mirroring. This has been an a priori choice to prevent any influence of flow disturbances close to the inlet and should not significantly alter the targeted metrics. A case with direct contact of solid phase and inflow boundary is considered in the next section. The resulting flow geometry for the current test case is depicted in Figure 5.6.

The material properties of fluid and solid phase as well as the particle size and count have been chosen identical to the cited reference. Furthermore, the damped Hertzian contact model of the form (4.1.36) is employed with the same value of Young modulus, Poisson ratio and coefficient of restitution (used in (4.1.38)) as in the *CFDEM* tutorial code. All parameters are summarized in Table 5.4.

To create the initial bed, a dilute distribution of particles with random positions within the column and above the perforated plate was created. At this point, none of the particles were in contact with each other, the walls or the plate. Then a pure DEM simulation of settling under gravity is performed to obtain the initial dense configuration for the coupled simulation. The resulting bed is depicted in Figure 5.7a and has a length along the vertical axis of $L_\mathrm{b} = 0.0146\,\mathrm{m}$.

For the coupling we use the drag force expression by Di Felice (see Table 3.1) which is also available in the *CFDEM* coupling. We further employ an implicit treatment of drag in the fluid phase and resort to model II discussed in Section 3.3, i.e. the pressure gradient and viscous stresses are not averaged from the solid particles to act as a source term to the fluid phase. In fact, the viscous stress force has again been observed to be three orders of magnitude smaller than either of the other coupling forces and is neglected within this benchmark. For the averaging we use the above-mentioned kernel approach with mirroring and a bandwidth of $b = 2D_\mathrm{s}$, where the Gaussian is cut off at a distance of three times the bandwidth. In Figure 5.7b we have visualized the values of solid volume fraction which were calculated from the initial particle configuration.

For the fluid phase we have chosen a smoothing length of $h = 0.002\,\mathrm{m} = 2D_\mathrm{s}$, leading to approximately 145000 points. At the bottom boundary we prescribe a uniform upward



(a) Initial particle configuration with bed length L_b depicted.

(b) Solid volume fraction attached to fluid points from initial particle configuration.

Figure 5.7: Initial particle configuration and solid volume fraction in the FPM simulation.

velocity which linearly increases from $U = 0.002\,\mathrm{m/s}$ to $U = 0.038\,\mathrm{m/s}$ within the considered time interval of $t \in [0, 0.2]\mathrm{s}$. As for the remaining boundaries, we used slip conditions at the walls and further applied a zero Neumann condition for velocity at the outflow. While both, the hydrostatic and dynamic pressure, are set to a constant value at the outflow boundary, we need to revisit our earlier discussion on pressure boundary conditions with respect to the slip walls and inflow. Due to the perforated plate and chosen value of averaging bandwidth, the latter boundary can once more be treated without considering the coupling terms. However, the same is clearly not valid for the side walls of the column. Since we employ model II from Section 3.3 and treat drag implicitly, (5.1.13) becomes

$$\frac{\partial p_{\mathrm{hyd}}}{\partial n} = \rho_{\mathrm{f}}(\boldsymbol{n} \cdot \boldsymbol{g}) + \rho_{\mathrm{f}} B(\boldsymbol{n} \cdot \boldsymbol{u}_{\mathrm{B}}) \tag{5.2.1}$$

While B can now be different from zero, $\boldsymbol{u}_{\mathrm{B}}$ represents the solid phase velocity at side walls. For non-moving walls which are impenetrable to the solid phase, one may require $\boldsymbol{n} \cdot \boldsymbol{u}_{\mathrm{B}} = 0$, once more reducing (5.2.1) to its single-phase version (5.1.12). For the dynamic pressure an analogous extension of (5.1.15) leads to

$$\frac{\partial p_{\mathrm{dyn}}}{\partial n} = -\rho_{\mathrm{f}}\left(\boldsymbol{n} \cdot \frac{\mathrm{D}\boldsymbol{u}_{\mathrm{f}}}{\mathrm{D}t}\right) + \boldsymbol{n} \cdot (\boldsymbol{\nabla} \cdot \boldsymbol{\tau}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{f}})) - \rho_{\mathrm{f}} B(\boldsymbol{n} \cdot \boldsymbol{u}_{\mathrm{f}}) \tag{5.2.2}$$

We observe that both, the first and last term on the right-hand side of (5.2.2), vanish at the side walls. For the coupling term this is a direct consequence of the geometry being stationary and impermeable to the fluid. For the first term we note that, since we no longer resort to a no-slip condition, boundary points are allowed to move in tangential directions. For planar walls, the normal does however remain constant along their path, such that the first term on the right-hand side of (5.2.2) also vanishes. In case of the inflow boundary, the term clearly does not vanish and needs to be properly adapted to the prescribed velocity.

In addition to boundary conditions it is also worth discussing our choice of time steps. As before, to determine a suitable time step for the solid phase, it is necessary to consider the drag relaxation time. Taking the minimum possible volume fraction $\varepsilon_{\mathrm{f}}^{\mathrm{HCP}} \approx 0.2595$ and the maximum inflow velocity, the relaxation time for drag based on the Di Felice correlation is given by

$$t_{\mathrm{r}} \approx 1.96 \times 10^{-3}\,\mathrm{s} \tag{5.2.3}$$

In contrast to the previous test case we also need to consider the constraint imposed by the DEM contact model. To determine this upper bound, we follow Table 4.1 and first need to determine the contact duration. For the Hertzian model it is necessary to estimate the maximum collision velocity within the system in order to determine the minimum contact duration from (4.1.83). Since we only consider the startup of fluidization with little dynamic behavior in the solid phase, taking the maximum fluid inflow velocity as the maximum collision velocity provides us with a very safe estimate. Following this line of thought we calculate

$$t_\mathrm{d} = 2.94\left(\frac{15}{16}\frac{m_\mathrm{s}^*}{\mathrm{Y}^*\sqrt{R_\mathrm{s}^*}}\right)^{\frac{2}{5}}\left(\frac{1}{v_{\mathrm{s},ij}^{n,0}}\right)^{\frac{1}{5}} \geq 2.24\times 10^{-4}\,\mathrm{m/s} \tag{5.2.4}$$

Taking the proposed number of 15 DEM steps within the contact duration (see Table 4.1) finally leads to

$$\Delta t_\mathrm{s} \leq \frac{t_\mathrm{d}}{15} \approx 1.49\times 10^{-5}\,\mathrm{s} \tag{5.2.5}$$

Clearly, even the underestimated relaxation time above leads to an irrelevant constraint in light of the strict DEM time step restriction. Thus, it is reasonable to employ a time step of

$$\Delta t_\mathrm{s} = 1\times 10^{-5}\,\mathrm{s} \tag{5.2.6}$$

which incidentally also coincides with the time step from the *CFDEM* tutorial, without the derivation being given there. For both, the initial settling and coupled flow, we again use the semi-implicit Euler scheme described in Section 4.1.5. For the fluid phase, we only need to consider the CFL condition as a constraint for the time step. It is easily verified that due to the choice of smoothing length and maximum inflow velocity in our current setup this poses a rather weak restriction and we use $\Delta t_\mathrm{f} = 2\times 10^{-4}\,\mathrm{s}$ which allows for 20 substeps in the procedure we visualized in Figure 4.13.

With all simulation parameters being specified, we now want to discuss the aforementioned pressure prediction which we compare our numerical results against. In Section 3.4, specifically (3.4.13), we saw that under the assumption of spatially and temporally constant fluid velocity as well as identical forces onto particles in the monodisperse bed, the fluid momentum equation reduces to

$$0 = -\varepsilon_\mathrm{f}\boldsymbol{\nabla} p + \rho_\mathrm{f}\varepsilon_\mathrm{f}\boldsymbol{g} - \boldsymbol{F}_\mathrm{d} n_\mathrm{p} \tag{5.2.7}$$

We further rewrite this by replacing $\boldsymbol{F}_\mathrm{d}$ according to (3.4.22), using $\varepsilon_\mathrm{s} = n_\mathrm{p} V_\mathrm{s}$ and dividing by the fluid volume fraction (which we know is always nonzero) so that

$$0 = -\boldsymbol{\nabla} p + \rho_\mathrm{f}\boldsymbol{g} - \frac{\beta}{\varepsilon_\mathrm{f}^2}\boldsymbol{u}_\mathrm{sf} \tag{5.2.8}$$

On the other hand, in the fluidized state the particle momentum equation reduces to

$$0 = -\boldsymbol{\nabla} p V_\mathrm{s} + \frac{V_\mathrm{s}\beta}{\varepsilon_\mathrm{s}\varepsilon_\mathrm{f}}\boldsymbol{u}_\mathrm{sf} + m_\mathrm{s}\boldsymbol{g} \tag{5.2.9}$$

By combining (5.2.8), (5.2.9) and using the fact that gravity is opposite to the relative velocity we obtain

$$\varepsilon_\mathrm{s}(\rho_\mathrm{s} - \rho_\mathrm{f})g = \frac{\beta}{\varepsilon_\mathrm{f}^2}U \tag{5.2.10}$$

In order to simplify notations in the above equation and the following, we denote by U the magnitude of superficial velocity which further coincides with its z-component in the

considered case. It is important to realize that, while we were considering velocities within the bed in the above derivation, a simple argument of mass conservation reveals that the superficial velocity is constant along the flow direction. We actually think of U as the inflow velocity in most of the upcoming analysis. Getting back to the relation (5.2.10), we recall that β correlations are functions of relative velocity and volume fraction. Keeping in mind that (5.2.10) is only valid at minimum fluidization conditions, replacing β provides us with a relation between superficial velocity and volume fraction at minimum fluidization. However, as we have seen in Section 3.4, these expressions may be of rather complex form. It is therefore customary to use the form of β representing the Ergun equation

$$\beta = 150\frac{\varepsilon_\mathrm{s}^2\mu}{\varepsilon_\mathrm{f} D_\mathrm{s}^2} + 1.75\frac{\rho_\mathrm{f}\varepsilon_\mathrm{s}}{D_\mathrm{s}\varepsilon_\mathrm{f}}U \tag{5.2.11}$$

As we have noted during the earlier discussion of this correlation, the second term in the Ergun equation acts as a correction for large particle Reynolds numbers. However, in the simulation setup at hand, we are faced with rather small Reynolds numbers and it is therefore suitable and in line with common practice [55] to neglect the second term. If we use this simplification and plug the remainder of β into (5.2.10), we obtain a prediction for the minimum fluidization velocity U_mf which reads

$$U_\mathrm{mf} = \frac{D_\mathrm{s}^2(\rho_\mathrm{s}-\rho_\mathrm{f})g}{150\mu}\frac{\left(\varepsilon_\mathrm{f}^\mathrm{mf}\right)^3}{1-\varepsilon_\mathrm{f}^\mathrm{mf}} \tag{5.2.12}$$

Still, we only have a relation between two a priori unknown quantities. A possible estimate to predict U_mf is described by Gidaspow [55] with reference to Wen and Yu [163] and is given by

$$\frac{1-\varepsilon_\mathrm{f}^\mathrm{mf}}{\left(\varepsilon_\mathrm{f}^\mathrm{mf}\right)^3} \approx 11 \tag{5.2.13}$$

This eliminates the volume fraction from (5.2.12) and therefore provides an expression that can be evaluated for the minimum fluidization velocity. Plugging both, U_mf and $\varepsilon_\mathrm{f}^\mathrm{mf}$, back into the Ergun equation

$$\frac{\Delta p}{L_\mathrm{b}} = 150\frac{\mu}{D_\mathrm{s}^2}\frac{\varepsilon_\mathrm{s}^2}{\varepsilon_\mathrm{f}^3}U + 1.75\frac{\rho_\mathrm{f}}{D_\mathrm{s}}\frac{\varepsilon_\mathrm{s}}{\varepsilon_\mathrm{f}^3}U^2 \tag{5.2.14}$$

finally provides the desired pressure drop per bed length. With the presented problem parameters and using the combination of (5.2.12) and (5.2.13) we estimate the minimum fluidization velocity

$$U_\mathrm{mf} = \frac{D_\mathrm{s}^2(\rho_\mathrm{s}-\rho_\mathrm{f})g}{1650\mu} \approx 7.888\times10^{-3}\,\mathrm{m/s} \tag{5.2.15}$$

For the considered inflow velocities $U \leq 0.038\,\mathrm{m/s}$ we obtain $Re_\mathrm{p} < 0.26$ which is well within the range of validity for (5.2.12) specified as $Re_\mathrm{p} < 20$ by Gidaspow [55]. From (5.2.13) we further obtain an estimate of

$$\varepsilon_\mathrm{f}^\mathrm{mf} \approx 0.38283 \tag{5.2.16}$$

Plugging both, the predicted porosity and fluid velocity, into the Ergun equation (5.2.14) we obtain an approximation of the pressure gradient at minimum fluidization. To predict the pressure drop across the particle bed, we need to calculate the bed length along the fluid flow direction. Clearly, this should be chosen consistent with the volume fraction, i.e.

$$L_\mathrm{b} = \frac{N_\mathrm{s}V_\mathrm{s}}{A_\mathrm{c}\varepsilon_\mathrm{s}^\mathrm{mf}} \approx 0.0141\,\mathrm{m} \tag{5.2.17}$$

and is identical for the prediction in both, the original *CFDEM* setup and our altered geometry, since the channel cross-section A_c remains the same. With this length we finally obtain a pressure drop of

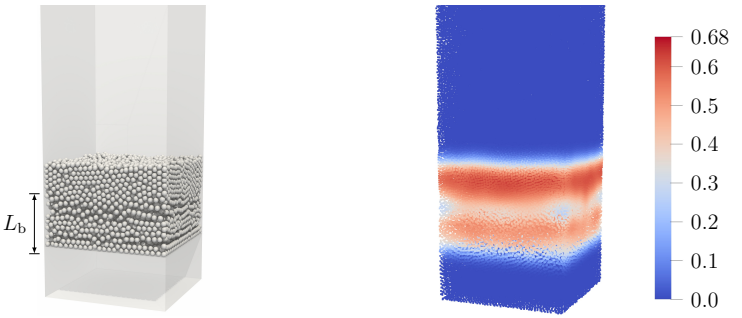
$$\triangle p^{\mathrm{WY}} \approx 169.76\,\mathrm{Pa} \tag{5.2.18}$$

We note that with the same volume fraction obtained from the approximation by Wen and Yu and the same bed length, one obtains a very similar pressure drop from the expression

$$\triangle p = L_b(1-\varepsilon_f^{mf})(\rho_s - \rho_f)g \approx 169.60\,\mathrm{Pa} \tag{5.2.19}$$

which can also be found in fluidization literature (cf. [36]). In the following we compare both, the results from our scheme and the *CFDEM* simulation, to the pressure drop at minimum fluidization as given in (5.2.18) and the velocity-dependent pressure drop before minimum fluidization based on (5.2.16) and the Ergun equation (5.2.14).

Simulation results



(a) Final particle configuration with initial bed length L_b depicted for comparison.

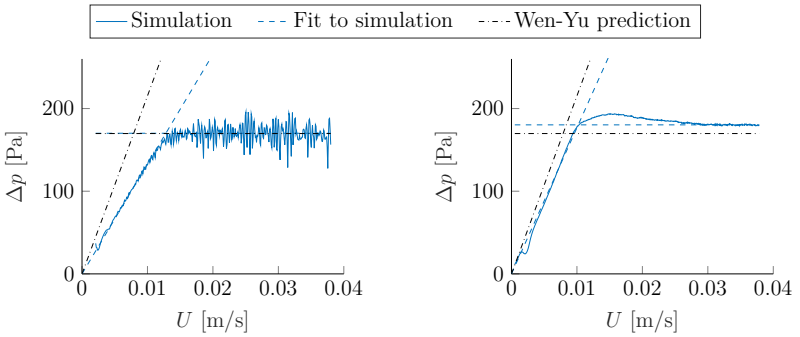
(b) Solid volume fraction attached to fluid points from final particle configuration.

Figure 5.8: Final particle configuration and solid volume fraction in the FPM simulation at $t = 0.2\,\mathrm{s}$.

Before we turn our attention towards this comparison, the reader may consult Figure 5.8 to gain an understanding of the change in particle configuration within the considered time frame $t \in [0, 0.2]$s. Namely, compared to Figure 5.7, the bed is only slightly lifted and a region of reduced solid volume fraction is formed in its center. Returning to the pressure drop, the results for $\Delta p = p_{in} - p_{out}$, i.e. the pressure difference between in- and outflow boundary, from the simulations with varying inflow velocity are shown in Figure 5.9. Due to the number of fluid points involved in our numerical simulation, not too much emphasis should be put on the difference in oscillating behavior, since the numerical resolutions differ significantly. The reason for this lies in a rather conservative a priori choice of h/b in order to ensure that enough points are within the support of the averaging kernel. In addition to the simulation results and the predicted value we have further added a linear and constant (mean) fit to both, the part before and after minimum fluidization, for better visual comparison to the prediction.

From Figure 5.9a we see that the pressure drop at minimum fluidization obtained from *LIGGGHTS* and *OpenFOAM* through the *CFDEM* coupling oscillates around the predicted value and reproduces it quite well in the mean. The pressure increase before minimum fluidization is reached does however deviate from the prediction based on the approximation

by Wen and Yu. This also holds true for the results obtained from our GFDM scheme, although the pressure increase is closer to the prediction. Comparing the pressure drop beyond the initial linear portion to both, the prediciton and *CFDEM* results, we observe that it exhibits an overshoot right after minimum fluidization which reduces to a constant pressure drop over time. Such overshoots are also in line with experimental observations [84]. The almost constant pressure drop of $\Delta p \approx 180.27\,\mathrm{Pa}$ towards the end of the simulation is however approximately 6% off in comparison to (5.2.18).



(a) Results of *CFDEM* simulation with $\Delta t_\mathrm{f} = 5 \times 10^{-4}\,\mathrm{s} = 50\Delta t_\mathrm{s}$ and 4608 cells.

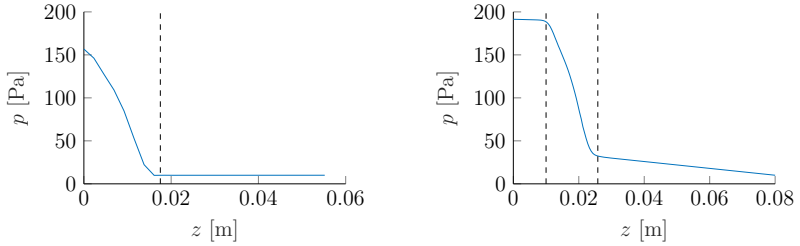
(b) Results of the FPM simulation with $\Delta t_\mathrm{f} = 2 \times 10^{-4}\,\mathrm{s} = 20\Delta t_\mathrm{s}$ and 145000 points.

Figure 5.9: Pressure drop over inflow velocity for both, the *CFDEM* and the FPM simulation.

To investigate the origin of the overpredicted static pressure drop, we investigated the pressure profile along the z-axis for both simulations. The profiles are visualized in Figure 5.10. For the *CFDEM* results the values of cells with identical z-coordinate have been averaged to produce the profile. For the FPM simulation we discretized the flow domain into slices of size $\Delta z = 0.0005m$ and took the average of values attached to points within each slice. The remaining parameters are the same as in Figure 5.9b. On the one hand we observe the expected pressure drop within the bed, marked by the dashed lines, on the other hand we do however notice a further linear decay towards the outflow boundary. This behavior is not present in the finite volume simulation which led to Figure 5.10a and may explain the fact that the pressure drop between in- and outflow is overpredicted by our simulation.

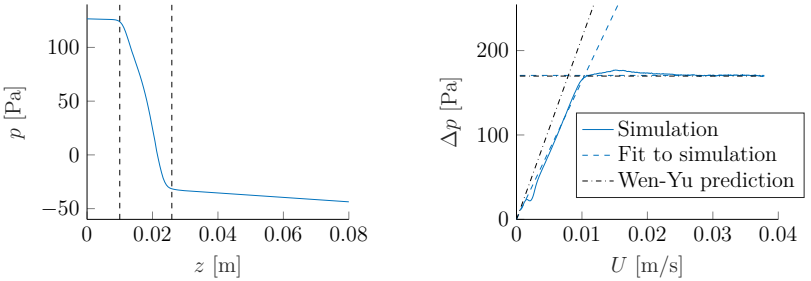
Based on this observation we chose to further investigate whether the expected profile can be enforced by specifying a zero gradient boundary condition for hydrostatic and dynamic pressure at the outflow. While this is numerically problematic, it is a physically more reasonable requirement since one typically does not observe truly uniform pressure at an outflow boundary. The results from a simulation with this altered setting are shown in Figure 5.11. We notice that prescribing the pressure gradient at the outflow is not sufficient to completely remove the post-bed linearity in pressure profile shown in Figure 5.11a. Nevertheless, the pressure drop across the fluidization column in Figure 5.11b reveals a significantly better agreement with the predicted value from (5.2.18). We now obtain a mean static pressure of $\Delta p \approx 170.56\,\mathrm{Pa}$ towards the end of the simulation, which is only approximately 0.5% above the value from (5.2.18).

While the pressure drop at minimum fluidization is now predicted very well, we still observe a fairly noticeable difference in slope for the linearly increasing pressure drop within the fixed bed. This is due to the fact that we used the very general approximation (5.2.13) instead



(a) Pressure profile from *CFDEM*. **(b)** Pressure profile from FPM.

Figure 5.10: Pressure profile along the z-axis at $t = 0.15\,\mathrm{s}$ and $U \approx 0.029\,\mathrm{m/s}$. The vertical dashed lines mark minimum and maximum z-coordinate of the bounding box which encloses all particles at the considered time. For the *CFDEM* tutorial, the minimum line is not shown since it is at $z = 0$.



(a) Pressure profile at $t = 0.15\,\mathrm{s}$ and $U \approx 0.029\,\mathrm{m/s}$. Plot setup identical to Figure 5.10b. **(b)** Pressure drop over inflow velocity. Plot setup identical to Figure 5.9b.

Figure 5.11: Results of FPM simulation with varying velocity and Neumann boundary condition for hydrostatic and dynamic pressure at the outflow.

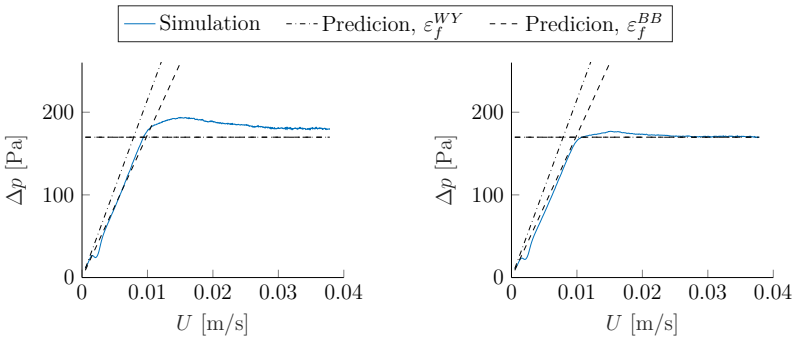
of tailoring towards our specific setup. To achieve the latter, we calculate the bounding box volume fraction from our initial particle configuration as

$$\varepsilon_{\mathrm{f}}^{\mathrm{BB}} = 1 - \frac{N_{\mathrm{s}} V_{\mathrm{s}}}{A_{\mathrm{c}} L_{\mathrm{b}}} \approx 0.4067 \tag{5.2.20}$$

where we used the previously stated bed length $L_{\mathrm{b}} = 0.0146\,\mathrm{m}$ and channel cross-sectional area $A_{\mathrm{c}} = w^2$. Plugging this volume fraction into (5.2.12) we obtain $U_{\mathrm{mf}} \approx 0.0098\,\mathrm{m/s}$, which in turn leads to a prediction of constant pressure drop from the Ergun equation of

$$\triangle p^{\mathrm{BB}} \approx 169.82\,\mathrm{Pa} \tag{5.2.21}$$

We observe that there is virtually no difference between this newly calculated value and the one from (5.2.18), meaning that our previous analysis remains valid. In Figure 5.12 we have again plotted the simulation results with Dirichlet and Neumann outflow boundary, this time together with both, the prediction based on (5.2.16) and (5.2.20). We observe that the linear part in both simulation results matches the Ergun equation very well, thus also validating the ability of our scheme to provide meaningful results before the onset of fluidization. We further note that using the fits from Figure 5.9b and Figure 5.11b, the



(a) Dirichlet boundary condition.

(b) Neumann boundary condition.

Figure 5.12: Pressure drop over inflow velocity from FPM simulation compared to predictions based on different volume fractions. Note that the fits to simulation data are not present any more and that the horizontal lines of the two predictions overlay each other.

minimum fluidization velocity from simulations is given by the intersection point of linear fit and mean value. Thus, we obtain values of $U_{\mathrm{mf}} \approx 0.0102\,\mathrm{m/s}$, $U_{\mathrm{mf}} \approx 0.0104\,\mathrm{m/s}$ for Dirichlet and Neumann boundary conditions, respectively. Also these values are fairly close to the prediction obtained by using the bounding box volume fraction.

Summary

This concludes our investigations for this specific test case. We observed that, independent of the outflow boundary condition, our scheme is able to capture the expected qualitative behavior in terms of pressure drop over inflow velocity. We further saw that the choice of a Neumann condition for hydrostatic and dynamic pressure led to results which provide a very good agreement in the pressure drop at minimum fluidization when compared to both, theory and the *CFDEM* code. In addition we were able to reproduce the expected pressure drop within the fixed bed before the onset of fluidization. While these results are promising, there is still room for future investigations which could not be conducted within the context of this work. In particular, this includes the linearity in pressure profile which is not observed in the finite volume code. Furthermore it would be interesting to investigate the influence of different combinations of numerical resolution and averaging bandwidths.

Since the current setting was quite different from the previous test case, in particular due to the number of particles not being insignificant any more, we again want to disclose the runtime of our simulations with the FPM. For both boundary conditions, the simulation took approximately 6 hours to complete. Due to the previously mentioned difference in spatial resolution, a comparison to the finite volume algorithm would be of little informative value and is thus omitted. To still put this into perspective, we note that a simulation with only the fluid phase, but the same spatial resolution and time step, takes approximately 64 minutes to complete. Clearly, in contrast to the previous test case, there is a significant increase in runtime due to the introduction of the solid phase. Much of this unsatisfactory scaling with particle number can currently be attributed to improper load balancing and MPI optimization and will be addressed in future work, when the numerical scheme is well-established and validated.

5.3 Spouted bed

After having considered a test case which showed a significant influence of the solid phase on the fluid, but with little motion within the particle bed, we now want to consider our last example which truly visualizes the combination of several dynamics within a coupled solid-fluid flow. To this end we consider the spouted bed setup which has originally been investigated, both experimentally and through simulations, by Link et al. [92]. In particular we chose "case A" from their article, which has since been investigated by many authors [3, 14, 34, 77, 90, 127] and provides us with the possibility of quantitative comparison. The geometry of this test case is depicted in Figure 5.13. In contrast to a fluidized bed, such as the one from the previous test case, the fluid, which is air in this case, does not enter the column with a uniform velocity across the inflow. Instead, the gas entering through a small slot in the center of the inflow does have a very large velocity while the gas entering at the sides acts as a low velocity background flow. This kind of spouted bed is also sometimes classified as spout-fluid bed (cf. [127]), while a classical spouted bed allows gas inflow only through the centered slot. Clearly, in such a setup the drag force experienced by particles is largest close to the center inlet and reduce both towards the outlet and along the x-axis. Therefore the bed exhibits a circulating behavior with particles being pushed upwards in the center region and falling downwards due to reduced drag close to the side walls (we always think of our viewpoint being aligned with the y-axis in Figure 5.13). One way to capture this dynamic in a measurement is to investigate the mass flux in vertical direction at several points along the x-axis and a fixed vertical position. This is exactly the quantity considered by Link et al. [92] and is the one used for quantitative comparison of our numerical simulations to existing CFD-DEM results.

Simulation setup

For our numerical simulations we use the geometry depicted in Figure 5.13 which is, aside form the height, identical to the one in the original work by Link et al. [92]. We further use the same combination of center inflow velocity $u_{\mathrm{f}}^{\mathrm{c}} = 30\,\mathrm{m/s}$ and velocity $u_{\mathrm{f}}^{\mathrm{s}} = 1.5\,\mathrm{m/s}$ at the side inlets. In accordance with the original work, the continuous medium is air, which we represent by the parameters $\rho_{\mathrm{f}} = 1.205\,\mathrm{kg/m^3}$ and $\mu = 1.813 \times 10^{-5}\,\mathrm{Pa\,s}$ corresponding to air at an ambient temperature of 20°C. The column is initially filled with $N_{\mathrm{s}} = 24500$ spherical solid particles of material density $\rho_{\mathrm{s}} = 2526\,\mathrm{kg/m^3}$ and diameter $D_{\mathrm{s}} = 2.5\,\mathrm{mm}$. As in the previous test case, this is done by randomly positioning particles so that they are neither in contact with each other nor the geometry. Then, a pure DEM simulation is performed to obtain the initial bed.

In the original article by Link et al. [92] the authors used a hard-sphere approach to model the particle interaction. To transfer their setup into our framework we thus need to determine suitable constants for our soft-sphere model. Here we chose to use the linear spring-dashpot model and determine constants based on the problem setup at hand. The most basic condition for the spring stiffness is obtained from the fact that contact model parameters need to be chosen so that they support static configurations of the particle such as the initial bed. A rather conservative choice is obtained by assuming that particles are perfectly stacked along the vertical direction and by requiring the spring stiffness in the lowermost particle to support the weight of all the above ones. With approximately H/D_{s} stacked particles each contributing a gravitational force $F_{\mathrm{g}} = m_{\mathrm{s}} g$ which needs to be counteracted by the spring force at maximum deformation $F_{\mathrm{el}} = k_{\mathrm{n}} \delta_{ij}^{\mathrm{n,max}}$, one obtains

$$k_{\mathrm{n}} \geq \frac{H}{D_{\mathrm{s}}} \frac{m_{\mathrm{s}} g}{\delta_{ij}^{\mathrm{n,max}}} \tag{5.3.1}$$

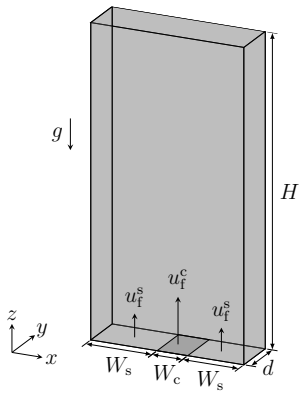


Figure 5.13: Geometry of spouted bed test case (not to scale).

	Symbol	Unit	Value
Center inlet size	W_c	mm	10
Side inlet size	W_s	mm	70
Cylinder depth	d	mm	15
Cylinder height	H	mm	1000
Gravity	g	m/s^2	9.81
Fluid density	ρ_f	kg/m^3	1.205
Dynamic viscosity	μ	Pa s	1.813×10^{-5}
Center velocity	u_f^c	m/s	30
Side velocity	u_f^s	m/s	1.5
Particle number	N_s	-	24 500
Particle diameter	D_s	mm	2.5
Solid density	ρ_s	kg/m^3	2526
Spring stiffness	k_n	N/m	1.2×10^4
Coeff. of Rest.	e_n	-	0.9

Table 5.5: Parameters of spouted bed test case.

While this might be sufficient for the initial settling, this lower bound is weaker than the one resulting from

$$k_n \geq \frac{m_s \left(v_{s,ij}^{n,0}\right)^2}{\left(\delta_{ij}^{n,max}\right)^2} \tag{5.3.2}$$

which we discussed in Section 4.1. Clearly, in the current setup the limiting factor is not due to velocities caused by gravitational acceleration, as is the assumption behind (4.1.16), but rather those caused by coupling forces. Most notably the drag force accelerates particles towards the fluid velocity. Thus, in the case of one-way coupling and constant fluid velocity throughout the flow domain, the maximum velocity due to coupling can be chosen as this background flow velocity. For the spouted bed setup presented above, one may choose the spout velocity. This certainly leads to a larger than necessary spring stiffness since this high velocity is only present in a small portion of the inflow and will quickly decrease throughout the column in a single-phase flow. However, including the obstruction due to the particle bed in our considerations, the fluid will maintain and possibly slightly overshoot the high velocity along the center line of the column. Since the particle acceleration due to drag is counteracted by gravity and collisions we do not expect the particle to exceed the spout velocity. This has also been verified by preliminary simulations such that we choose

$$k_n = \frac{m_s (u_f^c)^2}{\left(\delta_{ij}^{n,max}\right)^2} \tag{5.3.3}$$

With $\delta_{ij}^{n,max} = R_s$ this leads to $k_n = 1.2 \times 10^4$ N/m which we chose together with a damping constant derived from a coefficient of restitution $e_n = 0.9$ for our numerical simulation. As discussed in Section 4.1.5, the time step for a linear spring-dashpot model should be based on the critical time, which, for our parameters, evaluates to

$$t_c = \sqrt{\frac{m_s}{k_n}} \approx 4.15 \times 10^{-5}\,\text{s} \tag{5.3.4}$$

We further discussed that choosing $\Delta t_s = C t_c$ with $C = 0.2$ is often sufficient and that a very safe choice is given by $C = 0.1$. Within the initial settling we resorted to the rather

conservative choice of $\Delta t_\mathrm{s} = 4 \times 10^{-6}\,\mathrm{s}$. With these parameters the bounding box around the settled bed has a length in z-direction of $H_\mathrm{b} \approx 0.145\,15\,\mathrm{m}$.

In the previous test cases we have already discussed the pressure boundary conditions at length. Up until now we observed a reduction to their single phase version for the chosen setups. While this remains true for the side walls and outflow of the current setup, the attentive reader might have noticed from Figure 5.13 that no permeable plate has been added, so that the inflow acts as a wall to the particles and the bed settles onto this part of the geometry. Due to the configuration of different inflow regions, adding such a plate would have almost definitely skewed the results. Furthermore, having particles in direct contact with the inflow boundary now introduces a non-vanishing modification of the dynamic pressure condition and therefore gives us the opportunity to verify the correct treatment of such cases. This is due to the fact that $B \neq 0$ at the inflow and

$$\boldsymbol{n} \cdot \boldsymbol{u}_\mathrm{f} = \begin{cases} u_\mathrm{f}^\mathrm{c} & \text{at the center inlet} \\ u_\mathrm{f}^\mathrm{s} & \text{at the side inlet} \end{cases} \tag{5.3.5}$$

which means that the last term in (5.2.2) can not be crossed out. In early simulations we observed that omitting this fact led to unphysical results with a reversal in the direction of the total pressure gradient, which in turn prevented the bed of solid particles from being lifted. As for the remaining boundary conditions, we prescribed a uniform Dirichlet boundary condition for hydrostatic and dynamic pressure at the outflow, a slip condition for the velocity at side walls and Neumann zero condition for the outflow velocity.

As implied by the reference to (5.2.2), we again employ model II from Section 3.3 and treat drag implicitly. In contrast to the original work by Link et al. [92] we chose the more recent drag correlation by Rong et al. [125] which can be found in Table 3.1. In Section 3.4 we already stated that this correlation was derived from packed beds and for a large range of particle Reynolds numbers and volume fractions. Thus, it should be applicable in the current context, which includes both, dense regions which match the packed bed setting from [125] and dilute regions, where single particle drag modifications are most frequently applied. As we have discussed in Section 4.6, the drag force induces an additional time step restriction in the solid phase. To estimate this value we can calculate the bounding box volume fraction of the settled bed which is given by

$$\varepsilon_\mathrm{f}^{BB} = 1 - \frac{N_\mathrm{s} V_\mathrm{s}}{A_\mathrm{c} H_\mathrm{b}} \approx 0.386 \tag{5.3.6}$$

where A_c denotes the channel cross-section along the z-axis. Using this porosity and $u_\mathrm{f} = 30\,\mathrm{m/s}$ as the relative velocity in the drag correlation, we obtain a relaxation time of

$$t_\mathrm{r} \approx 1.26 \times 10^{-1}\,\mathrm{s} \tag{5.3.7}$$

for the above-mentioned Rong drag. We note that even for an unreasonably small volume fraction of $\varepsilon_\mathrm{f} \approx 0.3$, we would obtain $t_\mathrm{r} > 8 \times 10^{-2}\,\mathrm{s}$. Thus, the time step is clearly dictated by the DEM model and we chose to use $\Delta t_\mathrm{s} = 4 \times 10^{-6}\,\mathrm{s}$ throughout the simulation. For the fluid phase, on the other hand, we only need to fulfill the CFL condition, since drag is treated implicitly. For our simulations we chose a smoothing length of $h = 0.01\,\mathrm{m} = 4D_\mathrm{s}$ which lead to a discretization with $N_\mathrm{f} \approx 76\,000$ points. The CFL condition in the fluid phase is clearly determined by the center inflow velocity such that

$$\Delta t_\mathrm{f} \leq \mathcal{C} \frac{h}{u_\mathrm{f}^\mathrm{c}} \approx 6.67 \times 10^{-5}\,\mathrm{s} \tag{5.3.8}$$

for a choice of $\mathcal{C} = 0.2$. From this we inferred a suitable time step $\Delta t_\mathrm{f} = 6 \times 10^{-5}\,\mathrm{s}$, thus leading to 15 DEM substeps within each fluid time step, according to the procedure sketched

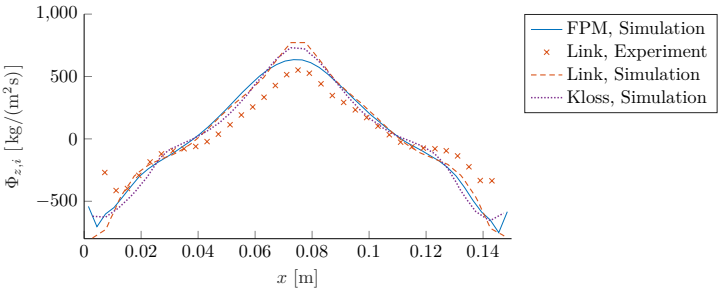


Figure 5.14: Time averaged solid phase mass flux across slices of the column cross-section at $z = 0.13\,\mathrm{m}$. Data points have been reconstructed from Figure 14 in [92] and Figure 13 in [77]. The mass flux was averaged over the time interval $t \in [4, 20]\mathrm{s}$.

in Section 4.7. We note that the maximum velocity within the fluid phase has further been monitored and we observed this choice to be sufficient to fulfill the CFL condition throughout the whole simulation time. Before we conclude this discussion and move on to the numerical results, we note that the coarse graining is again done with the mirroring discussed in Section 4.2. The Gaussian kernel bandwidth is chosen to be $b = 2D_\mathrm{s}$ and the weights are cut off at a distance of $2b$, which decreases computational cost and should not introduce any significant errors.

Quantitative evaluation

With all parameters of our simulation being clarified, we now return to the analysis of solid mass flux across the channel width, which originally motivated our choice of this specific test case. To measure this quantity, the channel cross-section at the same height as in the original work by Link et al. [92], i.e. $z_m = 0.13\,\mathrm{m}$, was discretized into 50 slices of width $\Delta x = 0.003\,\mathrm{m}$. Then, in each fluid time step from t^n to t^{n+1}, the set of particles $I_i^n \subset \{1, \dots, N_\mathrm{s}\}$, which passed through slice i during this time step, was determined. Each of these particles contributed to the mass flux via

$$\Phi_{z,i}^n = \frac{1}{\Delta t_\mathrm{f}} \sum_{j \in I_i^n} \mathrm{sign}(z_{\mathrm{s},j}^{n+1} - z_{\mathrm{s},j}^n) m_{\mathrm{s},j}, \quad i = 1, \dots, 50 \tag{5.3.9}$$

Finally the mass fluxes across a single slice were averaged over all time steps to obtain

$$\Phi_{z,i} = \frac{1}{N_t} \sum_{n=0}^{N_t - 1} \Phi_{z,i}^n \tag{5.3.10}$$

which is the quantity we compare to the previously reported results from [77, 92]. The value for each slice from our simulation was assigned to the x-coordinate $x_i = \frac{1}{2}\Delta x + (i-1)\Delta x$ and is plotted in Figure 5.14 together with the experimental and numerical data by Link et al. [92] and also the more recent simulation results by Kloss et al. [77]. Note that the numerical results in [77] were obtained from the *CFDEM* coupling discussed in the previous test case.

We observe that our simulation leads to a flux profile with a less pronounced peak and appears to be smoothed out when compared to both, the experimental and other numerical results. It does however provide a flux in the center region which is closer to the experimental findings and still agrees very well with the other numerical results towards the side walls. In

the following visual assessment of our results, we see that the upwards flow region towards the channel center does oscillate left and right, likely explaining the broadened peak. Another characteristic which is not shown to this extent by the previous numerical results, is the small decrease of flux close to the wall. This behavior can also be observed towards the left wall in the experimental results.

Overall, in terms of this metric, the results of our scheme look promising and are in good agreement with those reported in literature. We note that, in order to obtain an even better agreement with experimental results towards the side walls, a next step could be the inclusion of tangential forces. In particular, as has been discussed extensively in [58], the mass flux close to the walls can be decreased without significantly influencing the center region by including a rolling friction.

Qualitative evaluation

In addition to the mass flux profile we further want to provide a qualitative discussion of a series of snapshots which show the particle bed at different points in time. Since such a set of images has not been provided in the original work by Link et al., we draw comparisons to much more recent investigations by Saidi et al. [127] and Li et al. [90]. We note that both works are based on the *OpenFOAM* framework, similar to the *CFDEM* coupling, which was used in the previous test case, but with different implementations of the coupling and DEM scheme.

We begin with the start-up behavior, which occurs before a periodic motion is obtained. This portion of the simulation time is shown in Figure 5.15 and only present in [127]. The first row of snapshots shows the expected build-up of a single bubble at the center inlet and with little to no downward motion towards the side walls. It further agrees nicely with the snapshots up until $t = 100\,\text{ms}$ in [127]. In the second row of Figure 5.15, we observe that the bubble further grows and particles start to move downwards along the side walls, commencing a circulating behavior with particles again being pushed upwards by the center inflow. In this process the bed becomes increasingly thinned out above the initial bubble. This general behavior is in accordance with the findings in [127] up until about $t = 500\,\text{ms}$, although this process does take place within a smaller time frame in our simulation (up until $t = 330\,\text{ms}$). A characteristic which is not present in [127] is the build up of several smaller bubbles in the lifted bed above the initial bubble. A possible reason for this may be the fact that the authors in [127] considered an initially ordered bed, while we chose a random one. Small non-uniformities in our particle configuration may have been augmented over time, leading to the formation of additional bubbles. Another distinguishing factor is the formation of a bell-shaped dense region right above the center inlet while only isolated particles are distributed within the large bubble. In [127], the large bubble is instead filled more densely with particles and close to the inlet a V-shaped region of decreased solid volume fraction is formed. Nevertheless, we observe a further characteristic being shared between our results and those from [127], when we consider the collapse of the bubble in the last row of Figure 5.15. Namely, in the snapshot at $t = 500\,\text{ms}$ of [127], the bed, which is attained after the bubble collapsed, is already building up within the large bubble. The same can be said for the bell-shaped region which builds up towards the final snapshot at $t = 510\,\text{ms}$. In total we observed a lot of similarities within the start-up behavior when comparing our results to [127], but also noticed some differences. Possible sources for these will be discussed shortly.

This concludes our discussion on the spouted bed behavior during initialization. For the sake of completeness and for future comparisons we further provided a series of snapshots for the transitional regime after the collapse of the initial bubble and before the periodic

behavior commences. This can be found in Figure 5.16 and is not investigated in detail due to a lack of comparable snapshots in the cited sources.

If we consider the long-term behavior visualized by the snapshots in Figure 5.17, the most striking characteristic of our simulation is the fact that the bed periodically builds up close to either of the side walls with an almost identical slope of the bed surface. While the results in [127] also show deviations from a horizontal bed surface, a similar oscillating behavior can not be observed. The same can be said for the simulation in [90], which is also based on the Gidaspow drag law, i.e. the combination of Ergun and Wen-Yu drag (3.4.25), as in [127]. However, the authors in [90] conclude "that the Gidaspow drag closure produces unsatisfactory results for the spout-fluidization regime" and that both, the drag laws by Hoef et al. [65] and Beetstra et al. [7], are more appropriate. Their results based on these drag correlations do not show the angled bed surface which we observe in our simulation. In particular, for the drag by Hoef et al. the bed surface is almost plane in all of the provided snapshots.

Another noticeable characteristic is the shape of the high-velocity region within the bed. In our simulations this can be described as an S-shape which is periodically mirrored in the channel center. Since this S-shaped region always ends at the wall with increase bed height, it appears to be connected to the alternating bed slope. A less pronounced but similar shape can also be observed in [127] and, to an even lesser degree, in [90] with Gidaspow drag law. But a similar periodicity is absent in both cases. Furthermore, the two other drag laws tested in [90] show a rather straight upward flow in the channel center.

Lastly, we observe that, while bubble formation is present within our simulation, the bubbles are significantly smaller and the bed is much more dense than in both the cited sources. This clearly suggests that the upward forces, experienced by solid particles in our simulation are too low to loosen up the dense configuration, after the initial bubble, shown in Figure 5.15, has collapsed. Thus, the error is possibly rooted in either the pressure gradient force, the viscous stress force or the drag force. Firstly, it is common practice to neglect the viscous stress force in gas-solid fluidized beds, due to its negligible magnitude in comparison to the two other forces [26, 162]. This simplification has been adopted both, in our simulations and the above cited sources, so that we can exclude any influence of this term. On the other hand, none of the two articles includes a force based on the pressure gradient at particle locations. Instead, in [90] buoyancy is included by adjusting the density within the gravitational force. We note that this only coincides with our pressure gradient force when solely based on hydrostatic pressure and in case the coupling is not considered. In [127] this buoyancy modification is neglected, which we note is not a significant simplification compared to the model in [90] due to the ratio of solid to fluid density being very large. The fact that the results for Gidaspow drag in these two sources still compare better to each other than our simulation results, tells us that the pressure gradient force in our simulation may be too large in magnitude and cause the observed differences. While this term would thus be our prime suspect for further investigations, a lack of bed lift may also very likely be caused by an under-prediction of drag. Reasons for this could either be the choice of drag correlation, which deviates from the choices in both sources and has been shown to impact the results in [90], or could be due to errors within the calculation of volume fraction or the fluid phase velocity field.

Summary

This concludes the analysis of our final test case in the form of a spouted bed. We observed that our numerical scheme is capable of producing a good agreement in terms of solid mass flux profile, with both experimental and numerical results reported in literature. We also

found that many aspects of the start-up behavior agreed well with the results of a two-way coupled CFD-DEM simulation based on a finite volume scheme, which were presented in [127]. However, the long-term behavior determined by our scheme shows significant deviation from the cited sources and needs to be further investigated. For this we propose a number of steps that may be taken in future work to provide an analysis of this problem. Firstly, it is necessary to investigate whether using either the Gidaspow drag law or one of the two others used in [90], would significantly alter the results. Secondly, one may analyze the impact of including the pressure gradient forces versus considering only a buoyancy modification to gravity or even reducing the coupling forces to only drag. Lastly, we recall that no friction or rotation have been considered up to this point. It might very well be, that adding frictional forces can, for example, counteract the periodic surface build-up at the side walls.

In summary, we feel confident in stating that our numerical test cases have shown, that the unresolved coupling approach has been successfully integrated into our generalized finite difference framework. While certain observed aspects warrant a further analysis, the overall agreement with theoretical, experimental and previously reported numerical results, shows that the proposed algorithm is capable of reproducing typical phenomena of two-way coupled fluid-solid flows. Given more time and a thorough analysis of the mentioned aspects, it will be possible to clear out the remaining inconsistencies and open up the FPM framework to a lot of new fields of application.

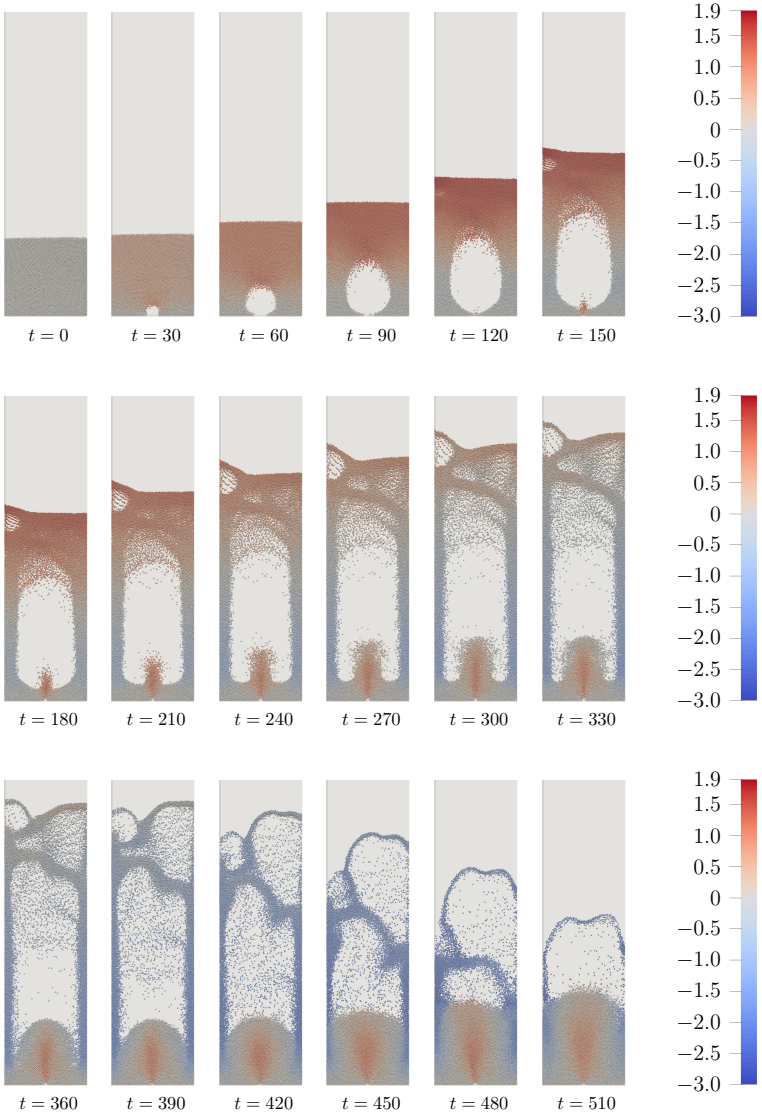


Figure 5.15: Build-up and collapse of bubble in spouted bed. Each solid particle is represented by a sphere glyph with its color corresponding to the velocity in z-direction in m/s. The time for each snapshot is given in ms. View direction is aligned with the y-axis. While the full width of the column is shown, the images have been cropped at a height of approximately $z = 0.57\,\mathrm{m}$. The fluid phase is not shown in these snapshots.

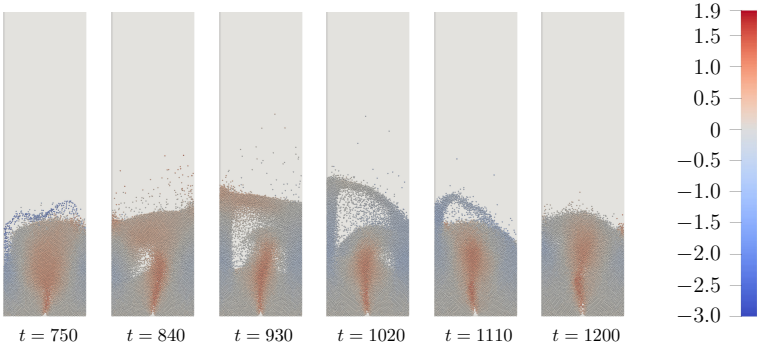


Figure 5.16: Build-up of a second small bubble after collapse of the initial bubble and before the periodic behavior starts. Each solid particle is represented by a sphere glyph with its color corresponding to the velocity in z-direction in m/s. The time for each snapshot is given in ms. View direction is aligned with the y-axis. While the full width of the column is shown, the images have been cropped at a height of approximately $z = 0.57\,\mathrm{m}$. The fluid phase is not shown in these snapshots.

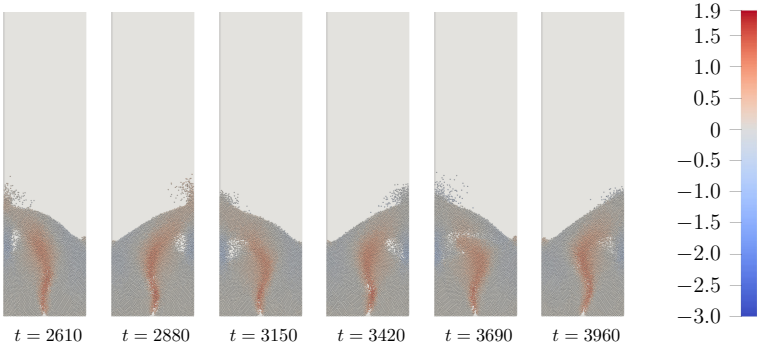


Figure 5.17: Periodic long-term behavior of the bed. Each solid particle is represented by a sphere glyph with its color corresponding to the velocity in z-direction in m s. The time for each snapshot is given in ms. View direction is aligned with the y-axis. While the full width of the column is shown, the images have been cropped at a height of approximately $z = 0.57\,\mathrm{m}$. The fluid phase is not shown in these snapshots.

Conclusion and outlook

In this work, we introduced a Lagrangian-Lagrangian two-way coupled CFD-DEM algorithm using the generalized finite difference framework of the FPM. We first provided a comprehensive introduction to the equations governing the fluid motion in a single-phase setting and to the generalized finite difference framework. On this basis, we discussed several methodologies that can be used to treat two-way coupled fluid-solid flows with a particulate solid phase and explained the choice of approach for this work. For the chosen unresolved CFD-DEM coupling we presented the theoretical background in the form of volume averaged Navier-Stokes (VANS) equations and discussed different formulations of the coupling found in literature as well as drag closures.

After providing details on the discrete element method (DEM) which was used to resolve the solid phase motion, we considered an aspect which, to the best of our knowledge, is novel in literature on unresolved CFD-DEM. Namely, we critically discussed how averaging schemes which are typically used in different numerical frameworks, such as finite volume methods or smoothed particle hydrodynamics, can be transferred to a Lagrangian GFDM framework. Within this discussion we carefully established a connection between the derivation of the VANS equations and conservation conditions which averaging schemes are typically constructed to fulfill. This showed that the weighted averaging with mirroring at boundaries of the flow domain provides fields which satisfy continuous conservation conditions and may be discretized in a manner consistent with the finite difference method. Many of the schemes proposed in literature are constructed to fulfill discrete versions of the conservation conditions, which necessitates a notion of volume attached to entities of the numerical discretization. In contrast to finite volume methods and meshfree schemes with mass points, such a notion of volume is not inherent in our framework. While we discussed approaches to define such volumes or circumvent the problem by introducing a second discretization, we concluded that the weighted averaging with mirroring is most consistent with the presented derivation of the VANS equations and our numerical scheme. This choice of scheme allows us to create well-founded tests which may later be used as a reference when considering different schemes.

We then extended the projection scheme used in FPM for fluid phases with constant density to incorporate the calculated volume fraction and coupling source terms. We emphasized how this sets itself apart from previous work in the context of the FPM and how it opens up the numerical framework to a wide range of previously inaccessible applications. After having discussed different time step restrictions and a substepping procedure in the solid phase routines, we validated the complete numerical scheme in three test cases of increasing complexity. In this context, we also discussed how boundary conditions need to be adapted so that they correctly account for the additional terms that result from a two-way coupling with the solid phase. While the test cases revealed some aspects which demand further investigation, an overall good agreement with analytical results, numerical results from other CFD-DEM schemes and experimental results was shown. In light of these observations we feel confident in stating that a well-founded basis for this type of coupling has been created in the considered generalized finite difference framework. Building upon this, we may extend the capabilities of our newly integrated coupling scheme in future work.

As the first among several next steps, we propose further analysis into the topic of coarse graining schemes in our GFDM framework. Since our coupling algorithm in conjunction with the weighted averaging and method of images has shown good results, we may now take this as a reference when analyzing averaging schemes which allow for more complex geometries. In this context it will be particularly interesting to consider free surfaces or moving boundaries, which form a typical field of application for Lagrangian fluid solvers. There we can directly build upon the discussion and ideas already presented in the current work. Additionally, one may analyze how different combinations of bandwidth and fluid phase resolution influence the result of the complete coupled simulation. In these extensions, emphasis should also be put on parallel performance which can be problematic due to the presence of three different sizes of neighborhoods resulting from the fluid phase, the DEM interactions and the coarse graining. With more investigations conducted on the importance of discrete conservation conditions of coarse graining in our framework, one may further analyze whether the proposed formulation of special stencils for the implicit drag treatment can provide any advantage over the formulations used in the presented test cases.

Clearly, future work will also include the extension of the DEM implementation. Specifically, we have seen that even in highly dynamic processes with continuous addition of kinetic energy, neglecting frictional forces may lead to non-negligible differences in macroscopic behavior. For both, these forces and the necessary rotational motion of particles, we already laid some of the groundwork in this thesis. With the rotations included we would also return to the discussion of integration schemes and expand upon the presented investigations also for translational motion.

Finally, there is an innumerable amount of phenomena in multiphase flow which we may want to represent in our scheme in the long-term. These can manifest itself in the necessity for additional coupling forces (e.g. virtual mass or lift), the integration of mass and heat transfer between phases or the treatment of non-constant density in the fluid phase.

As a concluding remark, we want to mention that in the process of working towards this thesis, significant time was also devoted to research on continuum models which aimed to capture the macroscopic behavior of a large number of interacting particles. Specifically, we extended an existing model based on mean field theory and purely elastic contacts, so that damping forces could be represented. Early results, part of which were published in [8], appeared to be promising. Therein we considered the motion under a confining potential and observed that damping from the microscopic contact model was captured reasonably well by the macroscopic equations. However, we found that for contact model parameters which are closer to those in a DEM simulation, the same setup lead to unsatisfactory results. Further removing the confining potential lead to unacceptable amounts of diffusion which are not present in the microscopic behavior. Therefore, we chose to omit a discussion of this topic within the current work, in favor of a more focused and well-rounded thesis. The interested reader may consult [8], where the initial investigations are described very concisely.

Appendices

A Tensor and vector calculus notations

As is common in the context of continuum mechanics, we made use of tensor calculus in many places within this thesis. In order to avoid any ambiguities, we devote this appendix to a clarification of the corresponding notation. We note that this closely follows the book by Spurk and Aksel [142], although minor modifications have been made to adapt their notation to our setting. All considerations are restricted to Cartesian coordinates with normalized basis vectors $\boldsymbol{e}_i$, $i \in \{1, \dots, d\}$. For our purposes we only need the following tensors:

- Zeroth order tensors in the form of scalar values $\phi \in \mathbb{R}$
- First order tensors $\boldsymbol{a} = a_i \boldsymbol{e}_i$ in the form of vectors $\boldsymbol{a} \in \mathbb{R}^d$
- Second order tensors $\mathbf{T} = \mathrm{T}^{(i,j)} \boldsymbol{e}_i \boldsymbol{e}_j$ in the form of matrices $\mathbf{T} \in \mathbb{R}^{d \times d}$

Note that we have used the Einstein summation convention in the above definitions. We start the introduction of operations on tensors by the *dyadic product* of two first order tensors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, which is given by

$$\boldsymbol{a} \otimes \boldsymbol{b} = a^{(i)} b^{(j)} \boldsymbol{e}_i \boldsymbol{e}_j$$

and results in a second rank tensor with components $\mathrm{T}^{(i,j)} = a^{(i)} b^{(j)}$. This product is not commutative, which is easily understood by thinking of the outer product as a product of column and row vector in the sense of

$$\boldsymbol{a} \otimes \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^T = \begin{bmatrix} a^{(1)} \\ \vdots \\ a^{(d)} \end{bmatrix} \begin{bmatrix} b^{(1)} & \dots & b^{(d)} \end{bmatrix} = \begin{bmatrix} a^{(1)} b^{(1)} & \dots & a^{(1)} b^{(d)} \\ \vdots & \ddots & \vdots \\ a^{(d)} b^{(1)} & \dots & a^{(d)} b^{(d)} \end{bmatrix}$$

From the matrix representation of this product we further observe that the identity

$$\boldsymbol{b} \otimes \boldsymbol{a} = (\boldsymbol{a} \otimes \boldsymbol{b})^T$$

holds. In addition to the dyadic product of two first rank tensors, we also want to consider products of first and second rank tensors, which take the form

$$\mathbf{T} \cdot \boldsymbol{a} = \mathrm{T}^{(i,j)} a^{(j)} \boldsymbol{e}^{(i)}$$
$$\boldsymbol{a} \cdot \mathbf{T} = \mathrm{T}^{(j,i)} a^{(j)} \boldsymbol{e}^{(i)}$$

These definitions are in line with common conventions for matrix-vector products and we therefore often neglect the dot symbol. For two first rank tensors, the product reduces to the inner product

$$\boldsymbol{a} \cdot \boldsymbol{b} = a^{(i)} b^{(i)} \in \mathbb{R}$$

for which we typically keep the dot symbol. By combining this with the above dyadic product, one can easily verify that the identity

$$(\boldsymbol{a} \cdot \boldsymbol{b}) \boldsymbol{b} = (\boldsymbol{b} \otimes \boldsymbol{b}) \boldsymbol{a}$$

holds. As the last kind of tensor operation we want to introduce

$$\mathbf{T} : \mathbf{S} = \mathrm{T}^{(i,j)}\mathrm{S}^{(i,j)} \in \mathbb{R}$$

which reduces two second rank tensors to a scalar and is called *contraction*.

Derivative notations

With this formalism being clarified, we want to take a look at how the partial derivatives, which inevitably arise in the continuum mechanical descriptions, are embedded into it. For $\boldsymbol{x} \in \mathbb{R}^d$ we denote by

$$\boldsymbol{\nabla}_{\boldsymbol{x}} = \left[\frac{\partial}{\partial x^{(1)}} \cdots \frac{\partial}{\partial x^{(d)}}\right]^T \tag{A.1}$$

the so-called *del operator*. When the argument of differentiation is clear, we usually omit the index. In agreement with common notation in continuum mechanics, we use this to define the *gradient* of a scalar and vector field through the dyadic product with the del operator. For a vector field $\boldsymbol{v} : \mathbb{R}^d \mapsto \mathbb{R}^d$ this reads

$$\boldsymbol{\nabla}\boldsymbol{v} \coloneqq \boldsymbol{\nabla} \otimes \boldsymbol{v} = \frac{\partial v^{(j)}}{\partial x^{(i)}} \boldsymbol{e}_i \boldsymbol{e}_j = \begin{bmatrix} \frac{\partial v^{(1)}}{\partial x^{(1)}} & \cdots & \frac{\partial v^{(d)}}{\partial x^{(1)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial v^{(1)}}{\partial x^{(d)}} & \cdots & \frac{\partial v^{(d)}}{\partial x^{(d)}} \end{bmatrix} \tag{A.2}$$

We note that, with this definition, the *Jacobian* of $\boldsymbol{v}$ is given by $\mathbf{J}_{\boldsymbol{v}} = (\boldsymbol{\nabla}\boldsymbol{v})^T$, where once more the subscript might be omitted whenever this introduces no ambiguity. For a scalar field $\phi : \mathbb{R}^d \mapsto \mathbb{R}$ this leads to

$$\boldsymbol{\nabla}\phi = \frac{\partial \phi}{\partial x^{(i)}} \boldsymbol{e}_i = \begin{bmatrix} \frac{\partial \phi}{\partial x^{(1)}} & \cdots & \frac{\partial \phi}{\partial x^{(d)}} \end{bmatrix}^T \tag{A.3}$$

where the dyadic product becomes a scalar multiplication. Using the del operator in conjunction with the tensor product, we obtain the generalization of the well-known first rank divergence

$$\boldsymbol{\nabla} \cdot \mathbf{a} = \frac{\partial a^{(k)}}{\partial x^{(k)}} \tag{A.4}$$

to a second rank tensor

$$\boldsymbol{\nabla} \cdot \mathbf{T} = \frac{\partial \mathrm{T}^{(k,i)}}{\partial x^{(k)}} \boldsymbol{e}_i \tag{A.5}$$

This means, that the vector divergence is applied column-wise to the matrix representation of the second rank tensor. This is also consistent with common representation of the laplacian for a vector field

$$\nabla^2 \boldsymbol{v} = \begin{bmatrix} \frac{\partial}{\partial x^{(k)}} \frac{\partial v^{(1)}}{\partial x^{(k)}} \\ \vdots \\ \frac{\partial}{\partial x^{(k)}} \frac{\partial v^{(d)}}{\partial x^{(k)}} \end{bmatrix} = \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla}\boldsymbol{v}) \tag{A.6}$$

which means that the laplacian is applied to each component individually. Finally, we want to provide a list of useful identities which are based on the above notation and are applied at several points throughout this thesis:

$$\boldsymbol{\nabla}(\phi \boldsymbol{v}) = \boldsymbol{\nabla}\phi \otimes \boldsymbol{v} + \phi \boldsymbol{\nabla}\boldsymbol{v} \tag{A.7}$$

$$\boldsymbol{\nabla} \cdot (\boldsymbol{u} \otimes \boldsymbol{v}) = \boldsymbol{u} \cdot \boldsymbol{\nabla}\boldsymbol{v} + (\boldsymbol{\nabla} \cdot \boldsymbol{u})\boldsymbol{v} \tag{A.8}$$

$$\boldsymbol{\nabla} \cdot (\phi \mathbf{T}) = \boldsymbol{\nabla}\phi \cdot \mathbf{T} + \phi(\boldsymbol{\nabla} \cdot \mathbf{T}) \tag{A.9}$$

$$\boldsymbol{\nabla} \cdot (\mathbf{T} \cdot \boldsymbol{v}) = (\boldsymbol{\nabla} \cdot \mathbf{T}) \cdot \boldsymbol{v} + \mathbf{T} : \boldsymbol{\nabla}\boldsymbol{v} \tag{A.10}$$

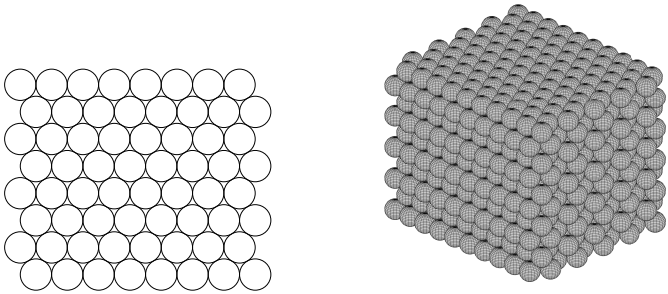
$$\boldsymbol{\nabla} \cdot (\boldsymbol{v} \cdot \mathbf{T}) = \left(\boldsymbol{\nabla} \cdot \mathbf{T}^T\right) \cdot \boldsymbol{v} + \mathbf{T}^T : \boldsymbol{\nabla} \boldsymbol{v} \tag{A.11}$$

$$\boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} \boldsymbol{u})^T = \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{u}) \tag{A.12}$$

$$\boldsymbol{\nabla} \cdot (\boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}) = \boldsymbol{u} \cdot \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{u}) + (\boldsymbol{\nabla} \boldsymbol{u})^T : \boldsymbol{\nabla} \boldsymbol{u} \tag{A.13}$$

$$\nabla^2 \boldsymbol{u} = \boldsymbol{\nabla} (\boldsymbol{\nabla} \cdot \boldsymbol{u}) - \boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \boldsymbol{u}) \tag{A.14}$$

B Close packing of identical spheres



(a) Hexagonal close packing of identical circular particles in 2D.

(b) Hexagonal close packing of identical spheres in 3D.

Figure B.1: Particle configurations for the hexagonal close packing.

In several places within this work, both, in the analysis of empirical force expressions and evaluation of simulation results, we are faced with the task of assessing whether values of porosity ε_f are physically reasonable. In the following we want to discuss the most simple criterion in the form of an upper bound for ε_s and, for that matter, lower bound of ε_f. While it is not a priori clear that there is such an upper bound for every particle shape (think of cubes), it does exists for the specific case of a monodisperse solid phase with spherical particles. In fact, this bound in three spatial dimensions is the content of the so-called *Kepler conjecture* which has been postulated in the 17th century and only been formally proven rather recently [59]. The configuration of particles which leads to the corresponding volume fraction is the so-called *hexagonal close packing* (HCP) with

$$\varepsilon_\mathrm{s}^\mathrm{HCP} = \frac{\pi}{\sqrt{18}} \approx 0.7405 \tag{B.1}$$

To put this into context, a random configuration of particles typically does not exceed volume fractions of $\varepsilon_\mathrm{s} = 0.68$ [154]. There is also a two-dimensional analogue to the above Kepler conjecture of sphere packing known as Thue's theorem for circle packings. It states (see [15] and references therein) that the densest configuration of equal circles leads to a volume fraction of

$$\varepsilon_\mathrm{s}^\mathrm{HCP} = \frac{\pi}{\sqrt{12}} \approx 0.9069 \tag{B.2}$$

Again, the HCP provides a configuration with this maximum solid volume fraction. To construct the HCP in two and three spatial dimensions, one places the sphere centroids at positions

$$\boldsymbol{x}_{\mathrm{s},ij} = \begin{bmatrix} (2i + (j \mod 2)) \\ \sqrt{3}j \end{bmatrix} R_\mathrm{s} \tag{B.3}$$

in $\mathbb{R}^2$ and

$$\boldsymbol{x}_{\mathrm{s},ijk} = \begin{bmatrix} 2i + (j + k \mod 2) \\ \sqrt{3}(j + \frac{1}{3}(k \mod 2)) \\ \frac{2\sqrt{6}}{3}k \end{bmatrix} R_\mathrm{s} \tag{B.4}$$

in $\mathbb{R}^3$ where $i = 1, \ldots, N_x$, $j = 1, \ldots, N_y$, $k = 1, \ldots, N_z$ and R_s denotes the sphere radius. The corresponding configurations are visualized in Figure B.1a and Figure B.1b.

C Single-particle drag force modifications

In Section 3.4 we discussed the class of drag correlations of the form

$$\boldsymbol{F}_\mathrm{d} = \frac{1}{8} C_\mathrm{d} \rho_\mathrm{f} \pi D_\mathrm{s}^2 \|\boldsymbol{u}_\mathrm{sf}\| \boldsymbol{u}_\mathrm{sf} g(\varepsilon_\mathrm{f}) \tag{C.1}$$

which represent modifications of the drag force onto an isolated spherical particle. We recall that these modifications are made to account for the presence of other particles. Within this appendix we want to provide the reader with a quantification of this modification. This is helpful in estimating the magnitude of forces that have to be expected in a coupled fluid-solid flow with particle configurations of varying volume fractions and for different Reynolds numbers. As before, we consider the voidage function

$$g(\varepsilon_\mathrm{f}) = \varepsilon_\mathrm{f}^{-\chi} \tag{C.2}$$

with exponents χ as presented in Section 3.4 and summarized in Table 3.1. We further restrict ourselves to the Dallavalle drag coefficient as proposed by Epstein (cf. (3.4.21))

$$C_\mathrm{d}(Re_\mathrm{p}) = \left(0.54 + \frac{4.8}{\sqrt{Re_\mathrm{p}}}\right)^2 \tag{C.3}$$

and note that the same results hold for the original coefficient of 0.63, as in (3.4.8). For reference, the drag on an isolated spherical particle, as given in (3.4.3), can be denoted by

$$\boldsymbol{F}_\mathrm{d}^0 = \frac{1}{8} C_\mathrm{d}^0 \rho_\mathrm{f} \pi D_\mathrm{s}^2 \|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\| (\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}) \tag{C.4}$$

where

$$C_\mathrm{d}^0 = C_\mathrm{d}\left(\frac{\rho_\mathrm{f}\|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\| D_\mathrm{s}}{\mu}\right) = C_\mathrm{d}\left(\frac{Re_\mathrm{p}}{\varepsilon_\mathrm{f}}\right) \tag{C.5}$$

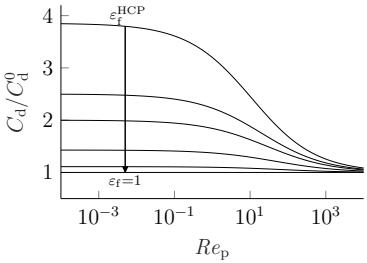
and

$$Re_\mathrm{p} = \frac{\varepsilon_\mathrm{f} \rho_\mathrm{f} \|\boldsymbol{u}_\mathrm{f} - \boldsymbol{v}_\mathrm{s}\| D_\mathrm{s}}{\mu} \tag{C.6}$$

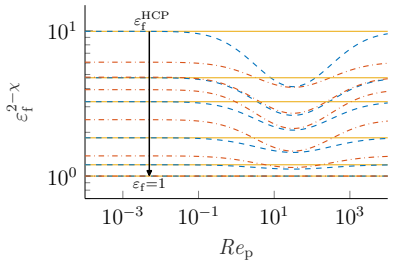
Upon comparing (C.1) and (C.4) we realize that

$$\boldsymbol{F}_\mathrm{d} = \boldsymbol{F}_\mathrm{d}^0 \frac{C_\mathrm{d}}{C_\mathrm{d}^0} \varepsilon_\mathrm{f}^{2-\chi} \tag{C.7}$$

such that the single particle drag is modified by a combination of change in drag force coefficient and a slightly altered voidage function. The magnitudes of both factors are visualized in Figure C.1a and Figure C.1b. We note that all fluid volume fractions have been chosen larger than the limit of densest sphere packing $\varepsilon_\mathrm{f}^\mathrm{HCP}$, which is described in Section B and forms a lower bound for physically reasonable values. As is to be expected, we observe from Figure C.1a that the drag coefficient C_d tends towards its single particle version C_d^0 with increasing fluid volume fraction and increasing Reynolds number. In the densest configuration, the coefficient is less than 4 times the size of C_d^0. On the other hand, from Figure C.1b we observe that the modification due to the exponential term can be as large as a factor of 10. This magnitude is however only attained for the Wen-Yu and Di Felice drag. Taking the more recent correlation by Rong et al., which is also proposed for a wide range of volume fractions, as a baseline, we observe that the former two correlations both likely overpredict the real drag force over a wide range of Reynolds numbers when a very dense configuration is considered. An exception to this is the range of medium Reynolds



(a) Modification of drag coefficient due to non-zero solid volume fraction.



(b) Voidage function contribution to drag force modification. The different drag correlations from Table 3.1 are distinguished by line style and color: Wen-Yu drag (solid, yellow), Di Felice (blue, dashed), Rong et al. (red, dash-dotted).

Figure C.1: Factors in the relative drag force $\boldsymbol{F}_d/\boldsymbol{F}_d^0$ plotted over a range of Reynolds numbers and for various volume fractions. The different lines of identical color represent the result for volume fractions $\varepsilon_f \in \{\varepsilon_f^{HCP}, 0.4, 0.5, 0.7, 0.9, 1.0\}$ from top to bottom (as visualized by the inset arrows).

numbers around $Re_p \approx 30$ where the Drag by Di Felice and Rong et al. are almost identical. These observations coincide with our discussions in Section 3.4, where we noted that the older drag force modifications were typically applied in dilute settings.

For the sake of completeness we also provide the complete drag modification, resulting from the combination of both factors, in Figure C.2. As is also evident from the individual plots, this plot shows that the drag in a dense configuration of monodisperse spherical particles and with one of the two older drag laws can be almost 40 times as large as the drag on an isolated particle.

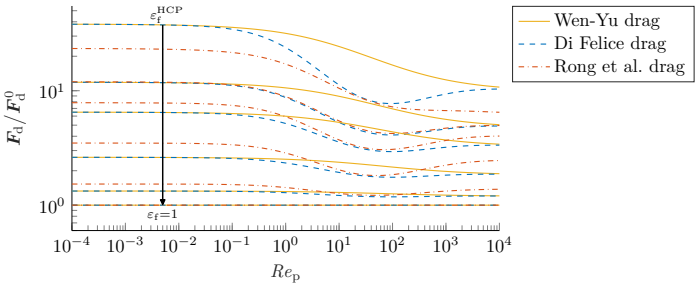


Figure C.2: Drag force modification due to non-zero solid volume fraction plotted over a range of Reynolds numbers and for various volume fractions. The different lines of identical color represent the result for volume fractions $\varepsilon_f \in \{\varepsilon_f^{HCP}, 0.4, 0.5, 0.7, 0.9, 1.0\}$ from top to bottom (as visualized by the inset arrow).

D Newton-Euler equations

In Section 4.1.1 we stated that the motion of a rigid spherical particle is described by the *Newton-Euler equations* which can be written as

$$m_\mathrm{s}\frac{\mathrm{d}\boldsymbol{v}_\mathrm{s}}{\mathrm{d}t} = \boldsymbol{F} \tag{D.1}$$

$$\mathrm{I}_\mathrm{s}\frac{\mathrm{d}\boldsymbol{\omega}_\mathrm{s}}{\mathrm{d}t} = \boldsymbol{T} \tag{D.2}$$

with translational and angular velocities given by

$$\frac{\mathrm{d}\boldsymbol{x}_\mathrm{s}}{\mathrm{d}t} = \boldsymbol{v}_\mathrm{s} \tag{D.3}$$

$$\frac{\mathrm{d}\boldsymbol{\theta}_\mathrm{s}}{\mathrm{d}t} = \mathbf{S}(\boldsymbol{\theta}_\mathrm{s})\boldsymbol{\omega}_\mathrm{s} \tag{D.4}$$

We did, however, not provide any derivation of these equations and left some details, such as the meaning of different coordinate systems or the form of $\mathbf{S}(\boldsymbol{\theta}_\mathrm{s})$, unconsidered. In this chapter we want to catch up on these things for the sake of completeness. In fact, we derive the more general form

$$\mathbf{I}\cdot\frac{d\boldsymbol{\omega}_\mathrm{s}}{dt} + \boldsymbol{\omega}_\mathrm{s}\times(\mathbf{I}\cdot\boldsymbol{\omega}_\mathrm{s}) = \boldsymbol{T} \tag{D.5}$$

for particles which need not be spherical. Our notation is mostly based on the lecture "Dynamics of mechanical multibody systems" held by Michael Burger at the university of Kaiserslautern. Good literature reference on this subject-matter are the books by Roberson and Schwertassek [123] and Wittenburg [167]. For a discussion of this topic in the context of the DEM we also refer to the book by Matuttis and Chen [103]. We clarify that this appendix mostly contains standard results of multibody mechanics and mainly serves to embed them into the notation of this thesis in order to avoid ambiguity. As we have already mentioned in Section 4.1.1, rotations have however been omitted in the simulations done for this thesis and presented in Chapter 5. Nevertheless, work towards their integration in our numerical scheme has already been done and this appendix both documents these efforts and serves as an outlook for future extensions. We note that, for the sake of notational convenience, we drop the solid phase index in the following.

D.1 Coordinate frames and velocity of material points

If we consider a rigid body which follows a translational and rotational motion, it is important to clarify the coordinate system in which we express its position and velocities. We distinguish between a so-called *body-fixed frame* $\mathcal{K}$, that has its origin in the body's center of mass and rotates with the body, and a non-moving frame in which we express the whole problem, the so-called *world frame* $\mathcal{I}$, also sometimes referred to as *inertial frame* or *laboratory coordinate system*.
At any given point in time we can express a vector $\boldsymbol{x}^\mathcal{I} \in \mathbb{R}^3$ in the inertial frame $\mathcal{I}$ by its coordinates in the body-fixed frame $\mathcal{K}$ and a rotation matrix $\mathbf{R}^{\mathcal{I}\mathcal{K}} \in SO(3)$, i.e.

$$\boldsymbol{x}^\mathcal{I} = \mathbf{R}^{\mathcal{I}\mathcal{K}}\boldsymbol{x}^\mathcal{K} \tag{D.6}$$

Above, we denoted by $SO(3)$ the group of rotations in $\mathbb{R}^3$ which can be identified with the group of orthogonal matrices in $\mathbb{R}^{3\times3}$ that have a determinant equal to 1. If we denote the vector from point P_1 to P_2 by $\boldsymbol{x}_{P_1P_2}$ and the origins of $\mathcal{I}$ and $\mathcal{K}$ by $0_\mathcal{I}$, $0_\mathcal{K}$, we observe (see

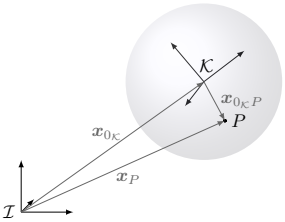


Figure D.1: Visualization of world frame, body-fixed frame and coordinates of a point P within a spherical particle.

Figure D.1) that the coordinates of any point P on the rigid body can be described with respect to the non-moving frame $\mathcal{I}$ through

$$\boldsymbol{x}^{\mathcal{I}}_{0_\mathcal{I}P} = \boldsymbol{x}^{\mathcal{I}}_{0_\mathcal{I}0_\mathcal{K}} + \mathbf{R}^{\mathcal{I}\mathcal{K}}\boldsymbol{x}^{\mathcal{K}}_{0_\mathcal{K}P} \tag{D.7}$$

Typically the position of a body is represented by the position of its center of mass, such that $\boldsymbol{x}^{\mathcal{I}}_{0_\mathcal{K}}$ coincides with the variable $\boldsymbol{x}_\mathrm{s}$ in our notation. For better readability we omit the index $0_\mathcal{I}$ in the following whenever a vector starting from the non-moving frame is considered. We observe that due to the rigid body assumption, any point within the body always has the same position in terms of the body-fixed coordinate system so that the vector $\boldsymbol{x}^{\mathcal{K}}_{0_\mathcal{K}P}$ is independent of time. By taking the time derivative of (D.7) we can thus conclude that the velocity with which point P moves in $\mathcal{I}$ is given by

$$\boldsymbol{v}^{\mathcal{I}}_p \coloneqq \dot{\boldsymbol{x}}^{\mathcal{I}}_p = \dot{\boldsymbol{x}}^{\mathcal{I}}_{0_\mathcal{K}} + \dot{\mathbf{R}}^{\mathcal{I}\mathcal{K}}\boldsymbol{x}^{\mathcal{K}}_{0_\mathcal{K}P} \tag{D.8}$$

Here we used the dot notation for time derivatives in order to maintain readability. Clearly, the first summand describes the translational velocity of the body's center of mass, denoted by $\boldsymbol{v}_\mathrm{s}$ in (D.1) and (D.3), while the second summand describes rotational velocities. Technically, we can directly store the 18 components of $\mathbf{R}^{\mathcal{I}\mathcal{K}} \in \mathbb{R}^{3\times3}$, $\dot{\mathbf{R}}^{\mathcal{I}\mathcal{K}} \in \mathbb{R}^{3\times3}$ for every particle to represent its orientation and rotational velocities, respectively. In this case we would have $\boldsymbol{\theta}_\mathrm{s} = \mathbf{R}^{\mathcal{I}\mathcal{K}}$, $\mathbf{S} = \mathbb{1}$ and $\boldsymbol{\omega}_\mathrm{s} = \dot{\mathbf{R}}^{\mathcal{I}\mathcal{K}}$ in (D.4). While this is a perfectly valid approach, upon investigation of the specific structure of $SO(3)$ it is revealed that much of the information is actually redundant and the number of variables which need to be stored can be significantly reduced. Since it will have a big impact on memory consumption for large systems of particles, we explore this in the following.

D.2 Angular velocity

The first reduction of variables we can make is within the components of rotational velocity. To do so, we define the *angular velocity matrix*

$$\mathbf{W}^{\mathcal{K}} \coloneqq (\mathbf{R}^{\mathcal{I}\mathcal{K}})^T\dot{\mathbf{R}}^{\mathcal{I}\mathcal{K}} \tag{D.9}$$

and write

$$\boldsymbol{v}^{\mathcal{I}}_p = \dot{\boldsymbol{x}}^{\mathcal{I}}_{0_\mathcal{K}} + \mathbf{R}^{\mathcal{I}\mathcal{K}}\mathbf{W}^{\mathcal{K}}\boldsymbol{x}^{\mathcal{K}}_{0_\mathcal{K}P} \tag{D.10}$$

From

$$(\mathbf{W}^{\mathcal{K}})^T + \mathbf{W}^{\mathcal{K}} = \frac{d}{dt}\left((\mathbf{R}^{\mathcal{I}\mathcal{K}})^T\mathbf{R}^{\mathcal{I}\mathcal{K}}\right) = \frac{d}{dt}\mathbb{1} = 0 \tag{D.11}$$

we see that the angular velocity matrix is skew-symmetric, such that we can write

$$\mathbf{W}^{\mathcal{K}} = \begin{bmatrix} 0 & -\omega_z^{\mathcal{K}} & \omega_y^{\mathcal{K}} \\ \omega_z^{\mathcal{K}} & 0 & -\omega_x^{\mathcal{K}} \\ -\omega_y^{\mathcal{K}} & \omega_x^{\mathcal{K}} & 0 \end{bmatrix} \tag{D.12}$$

By defining the *angular velocity vector* $\boldsymbol{\omega}^{\mathcal{K}} \coloneqq [\omega_x^{\mathcal{K}}\ \omega_y^{\mathcal{K}}\ \omega_z^{\mathcal{K}}]^T$ and using the correspondence between cross product and skew-symmetric matrices, equation (D.10) finally becomes

$$\boldsymbol{v}_p^{\mathcal{I}} = \dot{\boldsymbol{x}}_{0_{\mathcal{K}}}^{\mathcal{I}} + \mathbf{R}^{\mathcal{I}\mathcal{K}}(\boldsymbol{\omega}^{\mathcal{K}} \times \boldsymbol{x}_{0_{\mathcal{K}}P}^{\mathcal{K}}) \tag{D.13}$$

To obtain a formulation which is fully expressed in terms of the global coordinate system, we define

$$\mathbf{W}^{\mathcal{I}} \coloneqq \mathbf{R}^{\mathcal{I}\mathcal{K}}\mathbf{W}^{\mathcal{K}}(\mathbf{R}^{\mathcal{I}\mathcal{K}})^T = \dot{\mathbf{R}}^{\mathcal{I}\mathcal{K}}(\mathbf{R}^{\mathcal{I}\mathcal{K}})^T \tag{D.14}$$

and rewrite (D.10) as

$$\boldsymbol{v}_p^{\mathcal{I}} = \dot{\boldsymbol{x}}_{0_{\mathcal{K}}}^{\mathcal{I}} + \mathbf{W}^{\mathcal{I}}\boldsymbol{x}_{0_{\mathcal{K}}P}^{\mathcal{I}} \tag{D.15}$$

Since $\mathbf{W}^{\mathcal{I}}$ is again skew-symmetric, we also define $\boldsymbol{\omega}^{\mathcal{I}}$ as before and obtain

$$\boldsymbol{v}_p^{\mathcal{I}} = \dot{\boldsymbol{x}}_{0_{\mathcal{K}}}^{\mathcal{I}} + \boldsymbol{\omega}^{\mathcal{I}} \times \boldsymbol{x}_{0_{\mathcal{K}}P}^{\mathcal{I}} \tag{D.16}$$

We note that angular velocity vectors also have the expected relation

$$\boldsymbol{\omega}^{\mathcal{I}} = \mathbf{R}^{\mathcal{I}\mathcal{K}}\boldsymbol{\omega}^{\mathcal{K}} \tag{D.17}$$

which is easily seen from (D.13) by using the general identity $\mathbf{R}(\boldsymbol{a}\times\boldsymbol{b}) = (\mathbf{R}\boldsymbol{a})\times(\mathbf{R}\boldsymbol{b})$, valid for $\mathbf{R} \in SO(3)$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$. From this we observe that is sufficient to store either one of the angular velocity vectors $\boldsymbol{\omega}^{\mathcal{K}}, \boldsymbol{\omega}^{\mathcal{I}} \in \mathbb{R}^3$ instead of $\dot{\mathbf{R}}^{\mathcal{I}\mathcal{K}} \in \mathbb{R}^{3\times 3}$ to describe the velocity of material points within a DEM particle. Thus, what we denoted by $\boldsymbol{\omega}_{\mathrm{s}}$ in (D.2), (D.5) and (D.4) typically refer to one of these vectors. With the reduction of rotational velocity variables being clarified, we now want to proceed with a parametrization of the rotation matrix itself.

D.3 Parametrization of rotations in 3D

Common choices for the parametrization of the group of special orthoginal matrices $SO(3)$ are based on

- Compositions of three elemental rotations (e.g. Euler angles, Cardan angles)
- Four parameters representing an angle and axis of rotation (Euler parameters)

We begin with parameterizations based on three angle variables to see why an extension to four parameters might be useful.

Elemental rotations

As discussed in [123, 167], any matrix in $SO(3)$ can be represented by three successive elemental rotations, i.e. rotations about the axes of a coordinate system. Now suppose we have two frames $\mathcal{B}$ and $\mathcal{A}$ in $\mathbb{R}^3$ with basis vectors $\boldsymbol{e}^{\mathcal{B}} = [\boldsymbol{e}_1^{\mathcal{B}}\ \boldsymbol{e}_2^{\mathcal{B}}\ \boldsymbol{e}_3^{\mathcal{B}}]$, $\boldsymbol{e}^{\mathcal{A}} = [\boldsymbol{e}_1^{\mathcal{A}}\ \boldsymbol{e}_2^{\mathcal{A}}\ \boldsymbol{e}_3^{\mathcal{A}}]$ so that the former frame can be obtained by an elemental rotation around an axis of the latter one. If we denote this axis by $i \in \{1, 2, 3\}$, it is easy to check that we can write

$$\boldsymbol{e}^{\mathcal{B}} = \mathbf{R}^{\mathcal{B}\mathcal{A}}\boldsymbol{e}^{\mathcal{A}} = \mathbf{E}_i\boldsymbol{e}^{\mathcal{A}} \tag{D.18}$$

where $\mathbf{E}_i$ is one of the elemental rotation matrices

$$\mathbf{E}_1(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{bmatrix}, \quad \mathbf{E}_2(\psi) = \begin{bmatrix} \cos(\psi) & 0 & -\sin(\psi) \\ 0 & 1 & 0 \\ \sin(\psi) & 0 & \cos(\psi) \end{bmatrix}$$
$$\mathbf{E}_3(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

With these matrices and the knowledge that they can be composed to produce the body-fixed frame from the world frame, we write

$$\boldsymbol{e}^{\mathcal{K}} = \mathbf{E}_k \mathbf{E}_j \mathbf{E}_i \boldsymbol{e}^{\mathcal{I}} =: \mathbf{R}^{\mathcal{K}\mathcal{I}} \boldsymbol{e}^{\mathcal{I}} \tag{D.19}$$

with possibly different angles for every rotation. Consequently, the rotation matrix $\mathbf{R}^{\mathcal{I}\mathcal{K}}$, considered above, takes the form

$$\mathbf{R}^{\mathcal{I}\mathcal{K}} = (\mathbf{E}_i(\alpha))^T (\mathbf{E}_j(\beta))^T (\mathbf{E}_k(\gamma))^T \tag{D.20}$$

To define a specific parametrization of $SO(3)$ one defines the chosen sequence of rotation axes by fixing $i, j, k \in \{1, 2, 3\}$. Then, for a fixed sequence, each rotation matrix $\mathbf{R}^{\mathcal{I}\mathcal{K}}$ can be represented by the three angles α, β and γ which forms the orientation vector $\boldsymbol{\theta}_\mathrm{s} = [\alpha, \beta, \gamma]^T$ in (D.4). The choice of axes is however not completely arbitrary. It can be easily verified that choosing $i = j$ does not lead to independent rotations, but a single one with added angles. The same argument applies for the second and third rotation, such that the only allowed sequences are of the form

- $i = k$, $i \neq j$
- $i \neq j$, $i \neq k$, $j \neq k$

The first set of sequences is often referred to as *Euler angles* or *proper Euler angles*, while the second set is called *Tait-Bryan angles* or *Cardan angles*. It is customary to denote a specific sequence by $i - j - k$, i.e. define the rotation axis in order of appearance in (D.20). According to [123] and [167], the most commonly used Euler angle sequence is $3 - 1 - 3$. Sometimes also the specific angles to the sequence $1 - 2 - 3$ are called *Cardan angles* instead of the whole set of sequences. Let us list the two most important parametrization here:

- **Euler** $(3 - 1 - 3)$

$$\begin{aligned} \mathbf{R}^{\mathcal{I}\mathcal{K}}(\boldsymbol{\theta}_\mathrm{s}) &= (\mathbf{E}_3(\alpha))^T (\mathbf{E}_1(\beta))^T (\mathbf{E}_3(\gamma))^T \\ &= \begin{bmatrix} c\alpha c\gamma - s\alpha c\beta s\gamma & -s\alpha c\beta c\gamma - c\alpha s\gamma & s\alpha s\beta \\ s\alpha c\gamma + c\alpha c\beta s\gamma & c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha s\beta \\ s\beta s\gamma & s\beta c\gamma & c\beta \end{bmatrix} \end{aligned} \tag{D.21}$$

- **Cardan** $(1 - 2 - 3)$

$$\begin{aligned} \mathbf{R}^{\mathcal{I}\mathcal{K}}(\boldsymbol{\theta}_\mathrm{s}) &= (\mathbf{E}_1(\alpha))^T (\mathbf{E}_2(\beta))^T (\mathbf{E}_3(\gamma))^T \\ &= \begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta \\ c\alpha s\gamma + s\alpha s\beta c\gamma & c\alpha c\gamma - s\alpha s\beta s\gamma & -s\alpha c\beta \\ s\alpha s\gamma - c\alpha s\beta c\gamma & s\alpha c\gamma + c\alpha s\beta s\gamma & c\alpha c\beta \end{bmatrix} \end{aligned} \tag{D.22}$$

Note that we have abbreviated the trigonometric functions by their first letter to improve readability. With $\boldsymbol{\theta}_\mathrm{s}$ and $\mathbf{R}^{\mathcal{IK}}(\boldsymbol{\theta}_\mathrm{s})$ being properly defined, we now want to consider the matrix $\mathbf{S}(\boldsymbol{\theta}_\mathrm{s})$ which determines the relation between the temporal derivative of $\boldsymbol{\theta}_\mathrm{s}$ and the angular velocity vector $\boldsymbol{\omega}_\mathrm{s}$ in (D.20). Using $\mathbf{W}^{\mathcal{K}} = (\mathbf{R}^{\mathcal{IK}}(\boldsymbol{\theta}_\mathrm{s}))^T \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{R}^{\mathcal{IK}}(\boldsymbol{\theta}_\mathrm{s})\right)$, together with the chain rule in the temporal derivative and some lengthy but simple calculations, one obtains

$$\boldsymbol{\omega}^{\mathcal{K}} = \mathbf{T}^{\mathcal{K}}(\boldsymbol{\theta}_\mathrm{s})\dot{\boldsymbol{\theta}}_\mathrm{s}$$

with some matrix-valued function $\mathbf{T}(\boldsymbol{\theta}_\mathrm{s})$ depending on the kind of parametrization. Defining the inverse as $\mathbf{S}^{\mathcal{K}} \coloneqq \left(\mathbf{T}^{\mathcal{K}}\right)^{-1}$, we obtain the targeted formulation in (D.20) for the choice $\boldsymbol{\omega}_\mathrm{s} = \boldsymbol{\omega}^{\mathcal{K}}$. If instead we were to choose $\boldsymbol{\omega}_\mathrm{s} = \boldsymbol{\omega}^{\mathcal{I}}$, we observe from (D.17) that we can simply replace $\mathbf{S}^{\mathcal{K}}$ by $\mathbf{S}^{\mathcal{I}} = \mathbf{S}^{\mathcal{K}}(\mathbf{R}^{\mathcal{IK}})^T$. For the two above parameterizations we obtain

- **Euler** $(3-1-3)$

$$\mathbf{S}^{\mathcal{K}}(\boldsymbol{\theta}_\mathrm{s}) = \frac{1}{s\beta}\begin{bmatrix} s\gamma & c\gamma & 0 \\ s\beta c\gamma & -s\beta s\gamma & 0 \\ -c\beta s\gamma & -c\beta c\gamma & s\beta \end{bmatrix} \tag{D.23}$$

- **Cardan** $(1-2-3)$

$$\mathbf{S}^{\mathcal{K}}(\boldsymbol{\theta}_\mathrm{s}) = \frac{1}{c\beta}\begin{bmatrix} c\gamma & -s\gamma & 0 \\ c\beta s\gamma & c\beta c\gamma & 0 \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix} \tag{D.24}$$

We notice that both parameterizations lead to singularities in

$$\frac{\mathrm{d}\boldsymbol{\theta}_\mathrm{s}}{\mathrm{d}t} = \mathbf{S}^{\mathcal{K}}(\boldsymbol{\theta}_\mathrm{s})\boldsymbol{\omega}^{\mathcal{K}}$$

for certain angles β. Taking the Cardan angles as an example, the singularity at $\beta = \frac{n\pi}{2}$, $n \in \mathbb{N}$ results from the fact that the rotation matrix reduces to

$$\mathbf{R}^{\mathcal{IK}}(\boldsymbol{\theta}_\mathrm{s})\begin{bmatrix} 0 & 0 & 1 \\ \sin(\alpha+\gamma) & \cos(\alpha+\gamma) & 0 \\ -\cos(\alpha+\gamma) & \sin(\alpha+\gamma) & 0 \end{bmatrix} \tag{D.25}$$

such that all rotations of this type are described by β and a single parameter $\alpha+\gamma$, effectively reducing the degrees of freedom. This phenomenon is often referred to as as *gimbal lock*. It may be negligible for multibody systems with constrained motion but is problematic for discrete element simulations with freely rotating particles. This is why we want to consider a second type of parametrization based on four parameters and embedded in the quaternion formalism. This approach is able to efficiently circumvent the above problem by adding a fourth parameter.

Quaternions

Since we know that a rotation in 3D only has three degrees of freedom we need to impose an additional condition if we want to consider a parameter vector $\boldsymbol{q} \in \mathbb{R}^4$. This is reflected in the fact that we only consider normalized parameter vectors, i.e. $\|\boldsymbol{q}\| = 1$. To see how we can formulate a parametrization with these vectors, we take a closer look at the properties of our rotation matrix. We know that it is orthogonal and has a determinant of $+1$. Therefore all eigenvalues λ have a (complex) absolute value of $|\lambda| = 1$ and due to the odd dimension of $\mathbb{R}^3$ we know that at least one of them is $\lambda_1 = 1$. This tells us that the normalized eigenvector

$\boldsymbol{e}_1$ of λ_1 is invariant under the transformation $\mathbf{R}^{\mathcal{IK}}$, i.e. its coordinates are the same in both frames that the rotation maps between. In fact, every rotation matrix can be represented by its invariant eigenvector, which serves as rotation axis, and an angle of rotation around this axis. The angle can simply be calculated from the remaining eigenvalues. By setting

$$\cos(\phi) \coloneqq \frac{\operatorname{tr}\left(\mathbf{R}^{\mathcal{IK}}\right) - 1}{2} \tag{D.26}$$

it is easy to verify that they are given by

$$\lambda_{2/3} = \cos(\phi) \pm i \sin(\phi) = e^{\pm i\phi}$$

and that ϕ is the clockwise angle of rotation about the axis $\boldsymbol{e}_1$. With this it is customary to define $\boldsymbol{q} = (q_0, \tilde{q}) \in \mathbb{R}^4$ with

$$q_0 \coloneqq \cos\left(\frac{\phi}{2}\right), \quad \tilde{\boldsymbol{q}} \coloneqq \boldsymbol{e}_1 \sin\left(\frac{\phi}{2}\right) \tag{D.27}$$

which are often called *Euler parameters*. One can show (see [103, 123, 167], but beware that the authors present the transposed matrix) that the rotation matrix is then given by

$$\mathbf{R}^{\mathcal{IK}}(\boldsymbol{q}) = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \tag{D.28}$$

and by considering the trace and pairs of off-diagonal entries one can obtain the reverse relations

$$q_0 = \frac{1}{2}\sqrt{1 + \operatorname{tr}(\mathbf{R}^{\mathcal{IK}})} \tag{D.29}$$

$$q_1 = \frac{R_{32} - R_{23}}{4q_0}, \; q_2 = \frac{R_{13} - R_{31}}{4q_0}, \; q_3 = \frac{R_{21} - R_{12}}{4q_0} \tag{D.30}$$

Now, instead of using (D.28) to calculate the rotation matrix from the stored four component parameter vector $\boldsymbol{q} = (q_0, \tilde{q})$, it is customary to exploit the formalism of so-called *quaternions*. In the same way that classical complex numbers can be identified with two dimensional vectors, quaternions form a type of complex numbers with three imaginary components and can be identified with vectors $\boldsymbol{q} \in \mathbb{R}^4$. For our purpose it is only of relevance to know that the *quaternion product* of two vectors, or quaternions for that matter, $\boldsymbol{p} = (p_0, \tilde{p})$, $\boldsymbol{q} = (q_0, \tilde{q})$ is defined as

$$\boldsymbol{p} * \boldsymbol{q} = (p_0 q_0 - \tilde{\boldsymbol{p}} \cdot \tilde{\boldsymbol{q}}, p_0 \tilde{\boldsymbol{q}} + \tilde{\boldsymbol{p}} q_0 + \tilde{\boldsymbol{p}} \times \tilde{\boldsymbol{q}}) \tag{D.31}$$

One can further verify that this leads to a definition of inverse quaternion of the form

$$\boldsymbol{q}^{-1} = \frac{\overline{\boldsymbol{q}}}{\|\boldsymbol{q}\|^2} \tag{D.32}$$

with $\overline{\boldsymbol{q}} = (q_0, -\tilde{\boldsymbol{q}})$ denoting the *conjugate quaternion*. Obviously, as long as the quaternions are normalized, the definition of inverse and conjugate quaternion coincide. With these definitions one can show that

$$\boldsymbol{r}^{\mathcal{I}} = \mathbf{R}^{\mathcal{IK}}(\boldsymbol{q}) \boldsymbol{r}^{\mathcal{K}} \quad \Leftrightarrow \quad (0, \boldsymbol{r}^{\mathcal{I}}) = \boldsymbol{q} * (0, \boldsymbol{r}^{\mathcal{K}}) * \boldsymbol{q}^{-1} \tag{D.33}$$

so that the rotation matrix need not be constructed to rotate a vector. It is also easy to verify that the result of two successive rotations, i.e. $\boldsymbol{r}^{\mathcal{I}} = \mathbf{R}(\boldsymbol{q}_2)\mathbf{R}(\boldsymbol{q}_1)\boldsymbol{r}^{\mathcal{K}}$ can intuitively

be obtained by using $\boldsymbol{q} = \boldsymbol{q}_2 * \boldsymbol{q}_1$ in (D.33) and that replacing $\boldsymbol{q}$ by its conjugate in (D.33) provides the inverse mapping from world to body fixed coordinates.

As before, with a parameterization of the rotations being established, we want to examine the temporal derivative of the corresponding variables. Following [167, p.35] one can derive

$$\mathbf{S}^{\mathcal{K}}(\boldsymbol{q}) = \frac{1}{2}\begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix}^T, \quad \mathbf{S}^{\mathcal{I}}(\boldsymbol{q}) = \frac{1}{2}\begin{bmatrix} -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix}^T \tag{D.34}$$

such that

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \mathbf{S}^{\mathcal{K}}(\boldsymbol{q})\boldsymbol{\omega}^{\mathcal{K}} = \mathbf{S}^{\mathcal{I}}(\boldsymbol{q})\boldsymbol{\omega}^{\mathcal{I}}$$

which can also be written as

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \frac{1}{2}\boldsymbol{q} * (0, \boldsymbol{\omega}^{\mathcal{K}}) = \frac{1}{2}(0, \boldsymbol{\omega}^{\mathcal{I}}) * \boldsymbol{q} \tag{D.35}$$

This concludes our discussion on parametrizations of the rotation matrix. Next, we want to take a look at how forces acting on a rigid body induce the rotational motion we just described.

D.4 The result of applying forces and torques

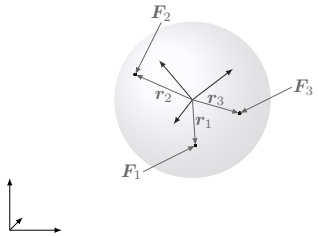


Figure D.2: Example of forces acting onto different points on a spherical particle.

For a body with mass m and translational velocity $\boldsymbol{v}_{0_{\mathcal{K}}}^{\mathcal{I}}$ the *linear* or *translational momentum* $\boldsymbol{P}^{\mathcal{I}}$ is defined as

$$P^{\mathcal{I}} = m \cdot \boldsymbol{v}_{0_{\mathcal{K}}}^{\mathcal{I}}$$

Analogously, a *angular* or *rotational momentum* $L^{\mathcal{I}}$ can be defined through

$$L^{\mathcal{I}} = \mathbf{I}^{\mathcal{I}} \cdot \boldsymbol{\omega}^{\mathcal{I}}$$

where the so-called *inertia tensor* $\mathbf{I}$ takes the role of the mass for the translational motion. The inertia tensor of a rigid body is defined in $\mathcal{K}$ because the rigidity implies that $\mathbf{I}^{\mathcal{K}}$ is constant. The relation between $\mathbf{I}^{\mathcal{I}}$ and $\mathbf{I}^{\mathcal{K}}$ is the same as for the angular velocity matrix, i.e.

$$\mathbf{I}^{\mathcal{I}} = \mathbf{R}^{\mathcal{I}\mathcal{K}}\mathbf{I}^{\mathcal{K}}(\mathbf{R}^{\mathcal{I}\mathcal{K}})^T \tag{D.36}$$

With these definitions the Newton-Euler equations now state that

- the change of linear momentum equals the sum of all forces acting on the body
- the change of angular momentum equals the sum of all torques acting on the body around the center of mass.

which translates into the equations

$$\dot{P}^{\mathcal{I}} = \sum_{j=1}^{n_F} \boldsymbol{F}_j^{\mathcal{I}} =: \boldsymbol{F}^{\mathcal{I}} \tag{D.37}$$

$$\dot{L}^{\mathcal{I}} = \sum_{j=1}^{n_T} \boldsymbol{T}_j^{\mathcal{I}} =: \boldsymbol{T}^{\mathcal{I}} \tag{D.38}$$

Note that to each force $\boldsymbol{F}_j$ on the body, having its point of attack not in the center of mass, we have a corresponding torque

$$\boldsymbol{T}_j^{\mathcal{I}} = \mathbf{R}^{\mathcal{I}\mathcal{K}} \boldsymbol{r}_j^{\mathcal{K}} \times \boldsymbol{F}_j^{\mathcal{I}} \tag{D.39}$$

with $\boldsymbol{r}_j$ being the vector from the origin of $\mathcal{K}$ to the point of attack of $\boldsymbol{F}_j$ (see Figure D.2). From (D.37) we directly obtain (D.1) by plugging in the definition of $P^{\mathcal{I}}$. For the angular momentum we have to be a bit careful because $\mathbf{I}^{\mathcal{I}}$, in contrast to the mass m, depends on the bodies movement and is therefore a function of time. By applying the chain rule we obtain

$$\dot{L}^{\mathcal{I}} = \dot{\mathbf{I}}^{\mathcal{I}} \boldsymbol{\omega}^{\mathcal{I}} + \mathbf{I}^{\mathcal{I}} \dot{\boldsymbol{\omega}}^{\mathcal{I}}$$

and observe that the second summand already has the form that appears in the second equation of (D.5). For the first summand one can use (D.36) and the fact that $\mathbf{I}^{\mathcal{K}}$ is constant to obtain

$$\dot{\mathbf{I}}^{\mathcal{I}} \boldsymbol{\omega}^{\mathcal{I}} = \mathbf{W}^{\mathcal{I}} \mathbf{I}^{\mathcal{I}} \boldsymbol{\omega}^{\mathcal{I}} - \mathbf{I}^{\mathcal{I}} \mathbf{W}^{\mathcal{I}} \boldsymbol{\omega}^{\mathcal{I}} = \boldsymbol{\omega}^{\mathcal{I}} \times (\mathbf{I}^{\mathcal{I}} \boldsymbol{\omega}^{\mathcal{I}}) \tag{D.40}$$

which completes (D.5). To obtain the simplified equation (D.2), we note that spherical particles have a scalar moment of inertia, i.e. $\mathbf{I}^{\mathcal{K}} = \mathbf{I}^{\mathcal{I}} = \mathrm{I_s} \mathbb{1}$, such that the above cross product vanishes.

E Derivations for the linear spring-dashpot model

In Section 4.1.2 we discussed the combination of a linear spring and damper force to model the contact between to DEM particles. During our discussion we stated several properties of this contact model without providing a derivation. This appendix is devoted to filling in some of these blanks.

E.1 Linear spring-dashpot model with attractive forces

Firstly, we consider the simple case where the damper is allowed to produce attractive forces within the separation phase. In this case we can easily derive the analytical solution for a binary collision. To do so, let us consider the equations of motion for two spherical particles moving only under the influence of the spring and damper force, i.e

$$\frac{\mathrm{d}\boldsymbol{x}_{\mathrm{s},i}}{\mathrm{d}t} = \boldsymbol{v}_{\mathrm{s},i}, \qquad m_{\mathrm{s},i}\frac{\mathrm{d}\boldsymbol{v}_{\mathrm{s},i}}{\mathrm{d}t} = \boldsymbol{F}_{\mathrm{c},ij} \tag{E.1}$$

$$\frac{\mathrm{d}\boldsymbol{x}_{\mathrm{s},j}}{\mathrm{d}t} = \boldsymbol{v}_{\mathrm{s},j}, \qquad m_{\mathrm{s},j}\frac{\mathrm{d}\boldsymbol{v}_{\mathrm{s},j}}{\mathrm{d}t} = \boldsymbol{F}_{\mathrm{c},ji} = -\boldsymbol{F}_{\mathrm{c},ij} \tag{E.2}$$

with

$$\boldsymbol{F}^n_{\mathrm{c},ij} = \left[k^*_{\mathrm{n}}\delta^n_{ij} - d_{\mathrm{n}}v^n_{\mathrm{s},ij}\right]\boldsymbol{n}_{ij} \tag{E.3}$$

The definitions of overlap δ^n_{ij}, normal velocity $v^n_{\mathrm{s},ij}$ and contact normal $\boldsymbol{n}_{ij}$ are identical to those in Section 4.1.2. As initial conditions we assume that the particles start with zero overlap and the relative velocity of the centroids upon collision aligns with the normal vector, i.e.

$$\delta^n_{ij}(t=0) = 0 \tag{E.4}$$

$$\boldsymbol{v}_{\mathrm{s},i}(t=0) - \boldsymbol{v}_{\mathrm{s},j}(t=0) = v^{n,0}_{\mathrm{s},ij}\boldsymbol{n}_{ij}(t=0) \tag{E.5}$$

We note that the latter assumption is also reasonable for different non-aligned relative velocities as long as the contact time is sufficiently short (cf. [134]). Taking the time derivative of the overlap we can easily see that

$$\frac{\mathrm{d}\delta^n_{ij}}{\mathrm{d}t} = -v^n_{\mathrm{s},ij} \tag{E.6}$$

such that

$$\frac{\mathrm{d}(\boldsymbol{v}_{\mathrm{s},i} - \boldsymbol{v}_{\mathrm{s},j})}{\mathrm{d}t} = \left(\frac{1}{m_{\mathrm{s},i}} + \frac{1}{m_{\mathrm{s},j}}\right)\left[k^*_{\mathrm{n}}\delta^n_{ij} + d_{\mathrm{n}}\frac{\mathrm{d}\delta^n_{ij}}{\mathrm{d}t}\right]\boldsymbol{n}_{ij} \tag{E.7}$$

It is evident that as long as the initial relative velocity is aligned with the normal vector, the contact normal does not change its direction during contact and we can take the scalar product of both sides with the normal to obtain the scalar equation

$$\frac{\mathrm{d}v^n_{\mathrm{s},ij}}{\mathrm{d}t} = \frac{1}{m^*_{\mathrm{s}}}\left[k^*_{\mathrm{n}}\delta^n_{ij} + d_{\mathrm{n}}\frac{\mathrm{d}\delta^n_{ij}}{\mathrm{d}t}\right] \tag{E.8}$$

Using (E.6) we can rewrite this as

$$m^*_{\mathrm{s}}\frac{\mathrm{d}^2\delta^n_{ij}}{\mathrm{d}t^2} - d_{\mathrm{n}}\frac{\mathrm{d}\delta^n_{ij}}{\mathrm{d}t} - k^*_{\mathrm{n}}\delta^n_{ij} = 0 \tag{E.9}$$

which is the classical second order ordinary differential equation describing a harmonic oscillator. By defining the *critical time*

$$t_{\mathrm{c}} = \sqrt{\frac{m^*_{\mathrm{s}}}{k^*_{\mathrm{n}}}} \tag{E.10}$$

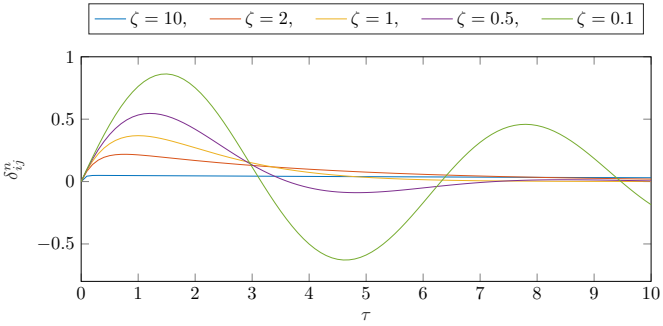


Figure E.1: Collision behavior for different damping ratios ζ_n (based on (E.14)).

we rewrite this ordinary differential equation (ODE) with respect to the dimensionless time $\tau = t/t_\mathrm{c}$, which leads to

$$\frac{\mathrm{d}^2\delta_{ij}^n}{\mathrm{d}\tau^2} - 2\zeta_\mathrm{n}\frac{\mathrm{d}\delta_{ij}^n}{\mathrm{d}\tau} - \delta_{ij}^n = 0 \tag{E.11}$$

with the so-called *damping ratio*

$$\zeta_\mathrm{n} = \frac{d_\mathrm{n}}{2\sqrt{m_\mathrm{s}^* k_\mathrm{n}^*}} \tag{E.12}$$

This ODE has the transient analytical solution

$$\delta_{ij}^n(\tau) = \begin{cases} e^{-\zeta_\mathrm{n}\tau}\left(c_1 e^{\tau\sqrt{\zeta_\mathrm{n}^2-1}} + c_2 e^{-\tau\sqrt{\zeta_\mathrm{n}^2-1}}\right) & \text{if } \zeta_\mathrm{n} > 1 \\ e^{-\tau}(c_1 + c_2\tau) & \text{if } \zeta_\mathrm{n} = 1 \\ e^{-\zeta_\mathrm{n}\tau}\left(c_1\cos\left(\sqrt{1-\zeta_\mathrm{n}^2}\tau\right) + c_2\sin\left(\sqrt{1-\zeta_\mathrm{n}^2}\tau\right)\right) & \text{if } \zeta_\mathrm{n} < 1 \end{cases} \tag{E.13}$$

where the cases describe overdamping, critical damping and underdamping from top to bottom respectively. Using the initial conditions to determine the coefficients c_1, c_2 we can derive

$$\delta_{ij}^n(\tau) = \tilde{v}_{\mathrm{s},ij}^{n,0} e^{-\zeta_\mathrm{n}\tau} \begin{cases} \dfrac{1}{\sqrt{\zeta_\mathrm{n}^2-1}}\sinh\left(\sqrt{\zeta_\mathrm{n}^2-1}\tau\right) & \text{if } \zeta_\mathrm{n} > 1 \\ \tau & \text{if } \zeta_\mathrm{n} = 1 \\ \dfrac{1}{\sqrt{1-\zeta_\mathrm{n}^2}}\sin\left(\sqrt{1-\zeta_n^2}\tau\right) & \text{if } \zeta_\mathrm{n} < 1 \end{cases} \tag{E.14}$$

where

$$\tilde{v}_{\mathrm{s},ij}^{n,0} = t_c v_{\mathrm{s},ij}^{n,0} \tag{E.15}$$

As can be seen in Figure E.1, for values of $\zeta_\mathrm{n} \geq 1$ there is no rebound of the particles. Instead, the overlap increases until a maximum is reached, after which it follows an exponential decay towards zero. This means that the complete kinetic energy is dissipated and the particles stick together indefinitely. Therefore the region of interest for granular flow with collisions of finite duration is $\zeta_\mathrm{n} \in (0, 1)$. Given a spring constant and desired damping behavior in terms of ζ_n, we can reformulate (E.12) to obtain the corresponding damper constant

$$d_\mathrm{n} = 2\zeta_\mathrm{n}\sqrt{m_\mathrm{s}^* k_\mathrm{n}^*} \tag{E.16}$$

Contact duration

Obviously, the collision in a DEM solver does not follow the oscillating behavior which occurs for $\zeta_\mathrm{n} < 1$. Instead, the particles separate when the overlap reaches zero again. From (E.14) it is easy to see that the period of oscillations is given by

$$\tau_\mathrm{d} = \frac{\pi}{\sqrt{1-\zeta_\mathrm{n}^2}} \tag{E.17}$$

Thus, the contact duration in terms of the original physical time scale is given by

$$t_\mathrm{d} = \frac{\pi}{\sqrt{1-\zeta_\mathrm{n}^2}} t_\mathrm{c} \tag{E.18}$$

We observe that in the purely elastic case, i.e. $\zeta_\mathrm{n} = 0$, the contact duration simplifies to

$$t_\mathrm{d} = \pi t_\mathrm{c} \tag{E.19}$$

Coefficient of restitution

As discussed in Section 4.1.2, it is more common in the context of DEM to use the so-called *coefficient of restitution* as a measure of the damping within a linear spring-dashpot model. This coefficient is defined as the ratio of velocity before and after the collision, i.e.

$$e_\mathrm{n} = \frac{\frac{\mathrm{d}\delta_{ij}^n}{\mathrm{d}\tau}(\tau = \tau_\mathrm{d})}{\tilde{v}_{\mathrm{s},ij}^{n,0}} \tag{E.20}$$

which provides a more direct means for calibration using experimental results. To calculate the numerator in this relation we note that the time derivative of the overlap is given by

$$\frac{\mathrm{d}\delta_{ij}^n}{\mathrm{d}\tau}(\tau) = \tilde{v}_{\mathrm{s},ij}^{n,0} \left[\cos\left(\sqrt{1-\zeta_\mathrm{n}^2}\tau\right) - \frac{\zeta_\mathrm{n}}{\sqrt{1-\zeta_\mathrm{n}^2}} \sin\left(\sqrt{1-\zeta_\mathrm{n}^2}\tau\right) \right] e^{-\zeta_\mathrm{n}\tau} \tag{E.21}$$

By plugging this into (E.20) we obtain

$$e_\mathrm{n} = e^{-\pi \frac{\zeta_\mathrm{n}}{\sqrt{1-\zeta_\mathrm{n}^2}}} \tag{E.22}$$

which shows the expected reduction of post-collisional velocity with increasing damping ratio. We now reformulate this result to obtain

$$\zeta_\mathrm{n} = -\frac{\ln(e_\mathrm{n})}{\sqrt{\pi^2 + \ln(e_\mathrm{n})^2}} \tag{E.23}$$

which is the relation commonly used in DEM with a linear spring-dashpot model to prescribe a specific damping behavior (cf. [109, 117]).

Maximum overlap and choice of spring stiffness

In Section 4.1.2 we provided the expression

$$k_\mathrm{n}^* \geq \frac{m_\mathrm{s}^* \left(v_{\mathrm{s},ij}^{n,0}\right)^2}{\left(\delta_{ij}^{\mathrm{n,max}}\right)^2} \tag{E.24}$$

for the spring stiffness which is necessary to prevent an overlap larger than $\delta_{ij}^{\mathrm{n,max}}$ for an initial collision velocity of $v_{\mathrm{s},ij}^{n,0}$. On the one hand, as noted in this section, one can derive this relation from a conservation of energy argument by noticing that

$$E = E_{\mathrm{kin}} = \frac{1}{2} m_{\mathrm{s}} \left(v_{\mathrm{s},ij}^{n,0} \right)^2 \tag{E.25}$$

at the beginning of the collision and

$$E = E_{\mathrm{pot}} = \frac{1}{2} k_{\mathrm{n}}^{*} \left(\delta_{ij}^{\mathrm{n,max}} \right)^2 \tag{E.26}$$

when the maximum overlap is obtained. On the other hand, one can verify this from the analytical solution by calculating the time at which the maximum overlap is obtained from

$$\frac{\mathrm{d}\delta_{ij}^{n}}{\mathrm{d}\tau}(\tau^{*}) = 0 \tag{E.27}$$

and (E.21). As is easily verified, the dimensionless time τ^{*} is given by

$$\tau^{*} = \frac{1}{\sqrt{1-\zeta_{\mathrm{n}}^{2}}} \cos^{-1}(\zeta_{\mathrm{n}}) \tag{E.28}$$

and by using this result in (E.14) we get

$$\delta_{ij}^{\mathrm{n,max}} = t_{\mathrm{c}} v_{\mathrm{s},ij}^{n,0} \exp\left(-\frac{\zeta_{\mathrm{n}}}{\sqrt{1-\zeta_{\mathrm{n}}^{2}}} \cos^{-1}(\zeta_{\mathrm{n}}) \right) \tag{E.29}$$

As is to be expected, we see that the maximum overlap takes the highest value for $\zeta_{\mathrm{n}} \to 0$, where

$$\delta_{ij}^{\mathrm{n,max}} = t_{\mathrm{c}} v_{\mathrm{s},ij}^{n,0} = \sqrt{\frac{m_{\mathrm{s}}^{*}}{k_{\mathrm{n}}^{*}}} v_{\mathrm{s},ij}^{n,0} \tag{E.30}$$

This can now be reformulated to obtain the same relation (E.24) as from the conservation of energy argument.

Another conclusion which can be drawn from (E.29) is the maximum collision velocity that can be handled by the contact model without having the particles pass through each other. By setting the overlap to the sum of particle radii, we see that this velocity satisfies

$$v_{\mathrm{s},ij}^{n,0} = \frac{R_{\mathrm{s},i} + R_{\mathrm{s},j}}{t_{\mathrm{c}}} \exp\left(\frac{\zeta_{\mathrm{n}}}{\sqrt{1-\zeta_{\mathrm{n}}^{2}}} \cos^{-1}(\zeta_{\mathrm{n}}) \right) \tag{E.31}$$

E.2 Linear spring-dashpot model without attractive forces

As we have noted when introducing the linear spring-dashpot model, during the unloading stage the damper might lead to attractive forces. It is common in DEM to set the contact force to zero once this point is reached. This does however mean that the analytical results derived in the previous subsection are no longer strictly valid. Neglecting the attractive forces means that we can expect the contact duration derived above to be overpredicted while the coefficient of restitution will be underpredicted. As a consequence the time step restriction based on the erroneous contact duration might not be strong enough and the desired damping behavior not represented accurately. The correct solution to the case without attractive

forces has been presented by Schwager and Pöschel [134] by adapting the conditions from which τ_d is derived. While in the above derivation one uses

$$\delta_{ij}^n(\tau_\mathrm{d}^*) = 0 \tag{E.32}$$

the correct end of contact is determined by

$$\left.\frac{\mathrm{d}^2\delta_{ij}^n}{\mathrm{d}\tau^2}\right|_{\tau_\mathrm{d}^*} = 0, \qquad \left.\frac{\mathrm{d}\delta_{ij}^n}{\mathrm{d}\tau}\right|_{\tau_\mathrm{d}^*} < 0 \tag{E.33}$$

and $\tau_\mathrm{d}^* > 0$. Taking the time derivative of (E.21) and setting it to zero we obtain the equation

$$\tan\left(\sqrt{1-\zeta_\mathrm{n}^2}\tau_\mathrm{d}^*\right) = -\frac{2\zeta_\mathrm{n}\sqrt{1-\zeta_\mathrm{n}^2}}{1-2\zeta_\mathrm{n}^2} \tag{E.34}$$

which is identical to the equation (20) in [134] but is reproduced here to avoid the cumbersome conversion between notations. As before, we only want to consider the solution of this equation for $\zeta_\mathrm{n} < 1$. We observe that the right-hand side is smaller than zero for $\zeta_\mathrm{n} < \frac{1}{\sqrt{2}}$ while it is larger for $\zeta_\mathrm{n} > \frac{1}{\sqrt{2}}$. Since τ_d^* needs to be positive this leads to the results

$$\tau_\mathrm{d}^* = \frac{1}{\sqrt{1-\zeta_\mathrm{n}^2}} \begin{cases} \pi - \operatorname{atan}\left(\dfrac{2\zeta_\mathrm{n}\sqrt{1-\zeta_\mathrm{n}^2}}{2\zeta_\mathrm{n}^2-1}\right) & \zeta_\mathrm{n} < \dfrac{1}{\sqrt{2}} \\ \operatorname{atan}\left(\dfrac{2\zeta_\mathrm{n}\sqrt{1-\zeta_\mathrm{n}^2}}{2\zeta_\mathrm{n}^2-1}\right) & \zeta_\mathrm{n} > \dfrac{1}{\sqrt{2}} \end{cases} \tag{E.35}$$

Plugging this dimensionless contact duration into (E.21) we obtain, in both cases,

$$\ln(e_\mathrm{n}) = -\zeta_\mathrm{n}\tau_\mathrm{d}^* \tag{E.36}$$

From (E.35) and (E.36) it is evident that the opposite expression of ζ_n in terms of the coefficient of restitution e_n can not be easily obtained. While the authors in [134] do not discuss this problem, Thornton et al. [152] propose a numerically fitted curve which they claim accurately reproduces the relation described above.

F Kernel mirroring and heat equation

In this appendix we want to provide more detail on the connection between a weighted averaging with Gaussian kernel function and the heat equation. To this end, we recall from standard PDE theory that solutions to the initial value problem

$$\frac{\partial \phi}{\partial t} - D\nabla^2 \phi = 0, \quad \phi(\boldsymbol{x}, t=0) = g(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d$$

of the heat equation on the full domain $\mathbb{R}^d$ can be obtained as a convolution

$$\phi(\boldsymbol{x}, t) = \int_{\mathbb{R}^d} \Phi_D(\boldsymbol{x} - \boldsymbol{y}, t) g(\boldsymbol{y}) \, \mathrm{d}V_y \tag{F.1}$$

of the *heat kernel*

$$\Phi_D(\boldsymbol{x}, t) = \begin{cases} \dfrac{1}{(4\pi D t)^{\frac{d}{2}}} e^{-\frac{\|\boldsymbol{x}\|^2}{4Dt}} & \text{if } t > 0 \\ 0 & \text{if } t < 0 \end{cases} \tag{F.2}$$

and the initial conditions $g \in \mathcal{C}\left(\mathbb{R}^d\right) \cap \mathcal{L}^\infty\left(\mathbb{R}^d\right)$ (cf. [45]). Note that the heat kernel is normalized on $\mathbb{R}^d$ for any $t \in (0, \infty)$, meaning that the above solution to the heat equation represents a volume averaging of the initial condition in the sense of Section 3.2. So in an unbounded domain, instead of averaging a field g with the normalized Gaussian weighting function

$$w_b(\boldsymbol{x}) = \frac{1}{(\pi b^2)^{\frac{d}{2}}} e^{-\frac{\|\boldsymbol{x}\|^2}{b^2}} \tag{F.3}$$

we can supply the field as initial condition to a heat equation and solve the equation until the time level

$$T = \frac{b^2}{4D} \tag{F.4}$$

is reached. Furthermore one can show that the method of images discussed in Section 4.2 can be applied to the heat kernel in order to provide a solution to the heat equation on the half-space bounded by a hyperplane (cf. [140]). The boundary condition fulfilled by

$$\phi(\boldsymbol{x}, t) = \int_{\mathbb{R}^d} [\Phi_D(\boldsymbol{x} - \boldsymbol{y}, t) + \Phi_D(\mathcal{M}_{\mathrm{w}}(\boldsymbol{x}) - \boldsymbol{y}, t)] g(\boldsymbol{y}) \, \mathrm{d}V_y \tag{F.5}$$

is the Neumann boundary condition

$$\left.\frac{\partial \phi}{\partial \boldsymbol{n}}\right|_{\boldsymbol{x}} = 0, \quad \forall \boldsymbol{x} \in \partial \mathcal{V}_\infty \tag{F.6}$$

This already suggests that quantities obtained by solving a heat equation on the bounded domain fulfill the conservation conditions which were restored by the mirroring. In fact, the heat equation with Neumann boundary condition is known to conserve the volume integral over the initial data. Thus, with properly chosen initial data, the conservation conditions are fulfilled by the analytical solution at any point in time. Upon comparing (F.1) and the definition of solid phase intrinsic volume averages

$$\varepsilon_{\mathrm{s}}(\boldsymbol{x}) \langle \psi \rangle^{\mathrm{s}}(\boldsymbol{x}) = \sum_{j=1}^{N_{\mathrm{s}}} \int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x} - \boldsymbol{y}) \psi(\boldsymbol{y}) \, \mathrm{d}V_y \tag{F.7}$$

we observe that the initial condition needs to be chosen as

$$g(\boldsymbol{x}) = \sum_{j=1}^{N_\mathrm{s}} \psi(\boldsymbol{x}) \chi_{\mathcal{V}_{\mathrm{s},j}}(\boldsymbol{x}) \tag{F.8}$$

to obtain

$$\Phi_D(\cdot, T) * g = \varepsilon_\mathrm{s} \langle \psi \rangle^\mathrm{s} \tag{F.9}$$

In the initial condition we denoted by $\chi_{\mathcal{V}_{\mathrm{s},j}}$ the indicator function within individual solid particle volumes. On the other hand, we can conclude from (F.1) and the particle phase average definition

$$n_\mathrm{p}(\boldsymbol{x}) \langle \psi \rangle^\mathrm{p}(\boldsymbol{x}) = \sum_{i=1}^{N_\mathrm{s}} w(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},i}) \psi_i \tag{F.10}$$

that

$$g(\boldsymbol{x}) = \sum_{j=1}^{N_\mathrm{s}} \psi_j \delta(\|\boldsymbol{x} - \boldsymbol{x}_{\mathrm{s},j}\|) \tag{F.11}$$

leads to

$$\Phi_D(\cdot, T) * g = n_\mathrm{p} \langle \psi \rangle^\mathrm{p} \tag{F.12}$$

Here we denoted by δ the Dirac delta function. Clearly, for an application of these results in the context of our coupled CFD-DEM scheme we still need to specify how they translate into a discretized setting and how the heat equation is solved numerically. The first application of a heat equation solver for the task of volume averaging and in the context of CFD-DEM can be traced back to Capecelatro et al. [14] as well as Sun and Xiao [148, 149]. In their work, the authors used a finite volume scheme which can typically guarantee that the fields resulting from a heat equation solver exactly fulfill the discrete conservation conditions. Unfortunately, in our framework we can only approximately satisfy the conditions and we need to take additional precautions to prevent unphysical undershoots in volume fractions. This is due to the fact that we can not guarantee positivity preserving stencils for the laplacian on circular neighborhoods in our generalized finite difference scheme [135, 144, 146]. Additionally, as discussed in Section 4.2.4, a special treatment of free surfaces would have to be devised. A thorough analysis of all these aspects and their influence on a CFD-DEM simulation would certainly be very interesting. In the context of the current work we did however not use this method of averaging and therefore leave these points open for future work. Before we conclude this appendix, we want to note that another problem in need of discussion in future work is how to represent the initial conditions. Clearly, the analytical description via indicator and delta functions is not be directly transferable to a discretized setting. While this has also been addressed in the original articles (cf. [14, 148, 149]), one should again be careful to what extent their treatment carries over to our framework.

G Approximation to weighting kernel integral

In this appendix we want to check the validity of the approximation

$$\int_{\mathcal{V}_{\mathrm{s},j}} w(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y \approx w(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},j})V_{\mathrm{s},j} \tag{G.1}$$

which we made in Section 4.2 to transition from volume averages, as defined in the derivation of the VANS equations, to discrete representations in the FPM framework. Specifically, we check this assumption for a Gaussian kernel

$$w_b(\boldsymbol{x}) = \tilde{w}_b(\|\boldsymbol{x}\|) = \frac{1}{(\pi b^2)^{\frac{d}{2}}} e^{-\frac{\|\boldsymbol{x}\|^2}{b^2}} \tag{G.2}$$

and spherical particles of equal radii, i.e $\mathcal{V}_{\mathrm{s},j} = B_{R_\mathrm{s}(\boldsymbol{x}_{\mathrm{s},j})}$, which represents the choice in all numerical simulations within the context of this work. To investigate the error introduced by the above approximation we use a numerical quadrature to calculate the integral on the left-hand side of (G.1). This is done by a simple discretization of the polar coordinate representation

$$\boldsymbol{y}_{kl} = \begin{bmatrix} \frac{r_k}{2}\cos\left(\frac{\phi_l}{2}\right) \\ \frac{r_k}{2}\sin\left(\frac{\phi_l}{2}\right) \end{bmatrix} \tag{G.3}$$

and spherical coordinate representation

$$\boldsymbol{y}_{klm} = \begin{bmatrix} \frac{r_k}{2}\sin\left(\frac{\theta_m}{2}\right)\cos\left(\frac{\phi_l}{2}\right) \\ \frac{r_k}{2}\sin\left(\frac{\theta_m}{2}\right)\sin\left(\frac{\phi_l}{2}\right) \\ \frac{r_k}{2}\cos\left(\frac{\theta_m}{2}\right) \end{bmatrix} \tag{G.4}$$

where

$$r_k = \frac{\Delta r}{2} + (k-1)\Delta r, \quad k \in \{1,\dots,N_r\}, \quad \Delta r = \frac{R_\mathrm{s}}{N_r} \tag{G.5}$$

$$\phi_l = \frac{\Delta \phi}{2} + (l-1)\Delta \phi, \quad l \in \{1,\dots,N_\phi\}, \quad \Delta \phi = \frac{2\pi}{N_\phi} \tag{G.6}$$

$$\theta_m = \frac{\Delta \theta}{2} + (l-1)\Delta \theta, \quad m \in \{1,\dots,N_\theta\}, \quad \Delta \theta = \frac{\pi}{N_\theta} \tag{G.7}$$

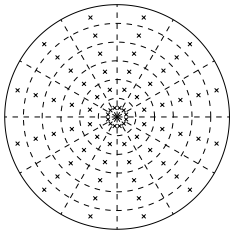
The area or volume for each of these quadrature points is then given by

$$A_{kl} = \frac{\Delta\phi}{2}\left[\left(r_k + \frac{\Delta r}{2}\right)^2 - \left(r_k - \frac{\Delta r}{2}\right)^2\right] \tag{G.8}$$

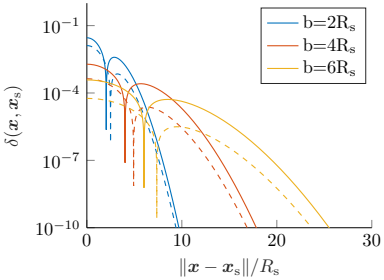
$$V_{klm} = \frac{\Delta\phi}{3}\left(\cos\left(\theta_m - \frac{\Delta\theta}{2}\right) - \cos\left(\theta_m + \frac{\Delta\theta}{2}\right)\right)\left[\left(r_k + \frac{\Delta r}{2}\right)^3 - \left(r_k - \frac{\Delta r}{2}\right)^3\right] \tag{G.9}$$

such that the integrals can be approximated using

$$\int_{B_{R_\mathrm{s}}(0)} w_b(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}y \approx \begin{cases} \sum_{k=1}^{N_r}\sum_{l=1}^{N_\phi} w_b(\boldsymbol{x}-\boldsymbol{y}_{kl})A_{kl} & d=2 \\ \sum_{k=1}^{N_r}\sum_{l=1}^{N_\phi}\sum_{m=1}^{N_\theta} w_b(\boldsymbol{x}-\boldsymbol{y}_{klm})V_{klm} & d=3 \end{cases} \tag{G.10}$$



(a) Discretization of a circular particle in two spatial dimension as used in the numerical integration. The points $\boldsymbol{y}_{kl}$ are marked by crosses.



(b) Error in the integral approximation for a single particle and different kernel bandwidths. The error is plotted over the relative distance to the particle centroid. Solid lines correspond to the result in $\mathbb{R}^2$ while dashed lines correspond to those in $\mathbb{R}^3$.

Figure G.1: Numerical setup for the kernel integration and resulting error for a single particle.

The discretization in $\mathbb{R}^2$ is visualized in Figure G.1a. At this point we start our investigation by denoting the error

$$\delta(\boldsymbol{x}, \boldsymbol{x}_\mathrm{s}) = \left| \int\limits_{B_{R_\mathrm{s}}(\boldsymbol{x}_\mathrm{s})} w_b(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y - w_b(\boldsymbol{x}-\boldsymbol{x}_\mathrm{s})V_\mathrm{s} \right| \tag{G.11}$$

which we calculate for a range of distances between the particle position $\boldsymbol{x}_\mathrm{s}$ and kernel center position $\boldsymbol{x}$. We note that due to

$$w_b(\boldsymbol{x}-\boldsymbol{x}_\mathrm{s})|B_{R_\mathrm{s}}| = w_{\frac{b}{R_\mathrm{s}}}\left(\frac{\boldsymbol{x}-\boldsymbol{x}_\mathrm{s}}{R_\mathrm{s}}\right)|B_1| \tag{G.12}$$

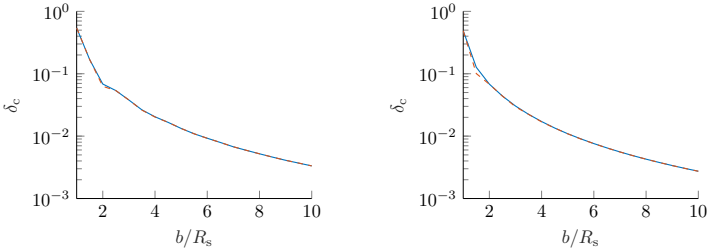
$$\int\limits_{B_{R_\mathrm{s}}(\boldsymbol{x}_\mathrm{s})} w_b(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}V_y = \int\limits_{B_1(\frac{\boldsymbol{x}-\boldsymbol{x}_\mathrm{s}}{R_\mathrm{s}})} w_b(R_\mathrm{s}\boldsymbol{y})R_\mathrm{s}^d\,\mathrm{d}V_y = \int\limits_{B_1(\frac{\boldsymbol{x}-\boldsymbol{x}_\mathrm{s}}{R_\mathrm{s}})} w_{\frac{b}{R_\mathrm{s}}}(\boldsymbol{y})\,\mathrm{d}V_y \tag{G.13}$$

the same errors are obtained at constant relative distance $\|\boldsymbol{x}-\boldsymbol{x}_\mathrm{s}\|/R_\mathrm{s}$ and relative bandwidth $\tilde{b} = b/R_\mathrm{s}$, independent of the particle radius. Thus, for our numerical investigation we can choose unit spheres without loss of generality. The results from a numerical integration with $N_r = 100$, $N_\phi = 360$, $N_\theta = 180$ and three different kernel bandwidths are shown in Figure G.1b and clearly reveal that the maximum error is obtained for $\boldsymbol{x}_\mathrm{s} = \boldsymbol{x}$. We further note that the local minimums correspond to the points where the distance equals the bandwidth. However, by itself, this result is of little use since it is not immediately clear which absolute error is acceptable within our context. Thus, to provide a better quantification of this error in a coupled simulation based on the volume averaged Navier-Stokes equations, we recall that solid volume fraction is defined as

$$\varepsilon_\mathrm{s}(\boldsymbol{x}) = \sum_{j=1}^{N_\mathrm{s}} \int\limits_{\mathcal{V}_{\mathrm{s},j}} w_b(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}y \tag{G.14}$$

We then denote the approximate volume fraction by

$$\tilde{\varepsilon}_\mathrm{s}(\boldsymbol{x}) = \sum_{j=1}^{N_\mathrm{s}} w_b(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{s},j})V_{\mathrm{s},j} \tag{G.15}$$



(a) Two spatial dimensions **(b)** Three spatial dimensions

Figure G.2: Cumulative error plotted over a range of relative bandwidths of the Gaussian kernel. For comparison the result of centering the kernel in a particle instead of taking the maximum of several offset positions is added as red dashed line.

so that we obtain the cumulative error

$$|\varepsilon_s(\boldsymbol{x}) - \tilde{\varepsilon}_s(\boldsymbol{x})| \leq \sum_{j=1}^{N_s} \delta(\boldsymbol{x}, \boldsymbol{x}_{s,j}) \leq \delta_c \tag{G.16}$$

made in the volume fraction due to the integral approximation. To estimate this cumulative error from errors made over a single particle volume in a robust way, we need to consider a dense configuration of particles. In fact, we choose the so-called hexagonal close packing (HCP) which has the highest volume fraction among all possible configurations, as described in Appendix B. To numerically calculate the cumulative error δ_c for a given relative kernel bandwidth $\tilde{b}$, we initially center the weighting function at a particle centroid location within the hexagonal close packing. Then we discard all particles of the HCP which have a distance larger than $c + 3R_s$ to that centroid, where

$$c = b \begin{cases} \sqrt{-\ln\left(\tilde{b}^2 \epsilon\right)} & d = 2 \\ \sqrt{-\ln\left(\dfrac{3\sqrt{\pi}}{4} \tilde{b}^3 \epsilon\right)} & d = 3 \end{cases} \tag{G.17}$$

is chose so that

$$\tilde{w}_b(c) V_s \leq \epsilon \tag{G.18}$$

for a value ϵ which is considered negligible and chosen to be $\epsilon = 1 \times 10^{-18}$ here. The increase of this cutoff distance by $2R_s$ is due to the fact that we offset the kernel within a radius of $2R_s$ and take the maximum of all cumulative errors among the offset positions. A further increase by R_s ensures that every particle which intersects the circle or sphere of radius c is fully considered in the calculation, thus leading to the above-stated $c + 3R_s$.

The results of our calculations are shown in Figure G.2a, Figure G.2b for two and three dimensions, respectively. We observe that choosing a bandwidth as small as the particle radius leads to a cumulative error which is clearly unacceptable. This is not surprising, since in this case the Gaussian kernel is far from constant within the extent of a single particle. On the other hand, for the range $b \in [4R_s, 8R_s]$ of typical values in CFD-DEM literature, we observe that the error is below 2% in both, 2D and 3D. We further have to take into account that we chose the densest possible particle configuration, meaning that the error has to be understood as an upper bound. In summary we conclude that (G.1) can be considered a valid approximation.

Bibliography

[1] J. Ai et al. "Assessment of rolling resistance models in discrete element simulations". In: *Powder Technology* 206.3 (2011), pp. 269–282. DOI: 10.1016/j.powtec.2010.09.030.

[2] M. Alian, F. Ein-Mozaffari, and S. R. Upreti. "Analysis of the mixing of solid particles in a plowshare mixer via discrete element method (DEM)". In: *Powder Technology* 274 (2015), pp. 77–87. DOI: 10.1016/j.powtec.2015.01.012.

[3] F. Alobaid, J. Ströhle, and B. Epple. "Extended CFD/DEM model for the simulation of circulating fluidized bed". In: *Advanced Powder Technology* 24.1 (2013), pp. 403–415. DOI: 10.1016/j.apt.2012.09.003.

[4] T. B. Anderson and R. Jackson. "Fluid Mechanical Description of Fluidized Beds. Equations of Motion". In: *Ind. Eng. Chem. Fund.* 6.4 (1967), pp. 527–539. DOI: 10.1021/i160024a007.

[5] D. Antypov and J. A. Elliott. "On an analytical solution for the damped Hertzian spring". In: *EPL (Europhysics Letters)* 94.5 (2011), p. 50004. DOI: 10.1209/0295-5075/94/50004.

[6] G. K. Batchelor. *An Introduction to Fluid Dynamics*. 1st ed. Cambridge University Press, 2000. DOI: 10.1017/CBO9780511800955.

[7] R. Beetstra, M. A. van der Hoef, and J. a. M. Kuipers. "Drag force of intermediate Reynolds number flow past mono- and bidisperse arrays of spheres". In: *AIChE Journal* 53.2 (2007), pp. 489–501. DOI: 10.1002/aic.11065.

[8] J. Bender, A. Klar, and J. Kuhnert. "Towards a unified Lagrangian multi-scale model for interacting particles". In: *PAMM* 18.1 (2018). DOI: 10.1002/pamm.201800209.

[9] J. J. Benito, F. Ureña, and L. Gavete. "Influence of several factors in the generalized finite difference method". In: *Applied Mathematical Modelling* 25.12 (2001), pp. 1039–1053. DOI: 10.1016/S0307-904X(01)00029-4.

[10] B. Blais et al. "Development of an unresolved CFD–DEM model for the flow of viscous suspensions and its application to solid–liquid mixing". In: *Journal of Computational Physics* 318 (2016), pp. 201–221. DOI: 10.1016/j.jcp.2016.05.008.

[11] C. M. Boyce et al. "Novel fluid grid and voidage calculation techniques for a discrete element model of a 3D cylindrical fluidized bed". In: *Computers & Chemical Engineering* 65 (2014), pp. 18–27. DOI: 10.1016/j.compchemeng.2014.02.019.

[12] N. V. Brilliantov et al. "Model for collisions in granular gases". In: *Physical Review E* 53.5 (1996), pp. 5382–5392. DOI: 10.1103/PhysRevE.53.5382.

[13] G. Buresti. "A note on Stokes' hypothesis". In: *Acta Mechanica* 226.10 (2015), pp. 3555–3559. DOI: 10.1007/s00707-015-1380-9.

[14] J. Capecelatro and O. Desjardins. "An Euler–Lagrange strategy for simulating particle-laden flows". In: *Journal of Computational Physics* 238 (2013), pp. 1–31. DOI: 10.1016/j.jcp.2012.12.015.

[15] H.-C. Chang and L.-C. Wang. *A Simple Proof of Thue's Theorem on Circle Packing*. 2010. arXiv: 1009.4322 [math.MG].

[16] S. Childress. *An introduction to theoretical fluid mechanics*. Courant lecture notes in mathematics 19. New York, NY: Courant Inst. of Math. Soc, 2009. ISBN: 978-0-8218-4888-3.

[17] A. J. Chorin. "Numerical solution of the Navier-Stokes equations". In: *Mathematics of Computation* 22.104 (1968), pp. 745–762. DOI: 10.1090/S0025-5718-1968-0242392-2.

[18] K. W. Chu et al. "CFD–DEM simulation of the gas–solid flow in a cyclone separator". In: *Chemical Engineering Science* 66.5 (2011), pp. 834–847. DOI: 10.1016/j.ces.2010.11.026.

[19] K. Chu et al. "CFD-DEM modelling of multiphase flow in dense medium cyclones". In: *Powder Technology* 193.3 (2009), pp. 235–247. DOI: 10.1016/j.powtec.2009.03.015.

[20] K. Chu, J. Chen, and A. Yu. "Applicability of a coarse-grained CFD–DEM model on dense medium cyclone". In: *Minerals Engineering* 90 (2016), pp. 43–54. DOI: https://doi.org/10.1016/j.mineng.2016.01.020.

[21] D. A. Clarke et al. "Investigation of Void Fraction Schemes for Use with CFD-DEM Simulations of Fluidized Beds". In: *Industrial & Engineering Chemistry Research* 57.8 (2018), pp. 3002–3013. DOI: 10.1021/acs.iecr.7b04638.

[22] P. W. Cleary. "Prediction of coupled particle and fluid flows using DEM and SPH". In: *Minerals Engineering*. Special issue: Comminution 73 (2015), pp. 85–99. DOI: 10.1016/j.mineng.2014.09.005.

[23] P. W. Cleary and R. D. Morrison. "Prediction of 3D slurry flow within the grinding chamber and discharge from a pilot scale SAG mill". In: *Minerals Engineering* 39 (2012), pp. 184–195. DOI: 10.1016/j.mineng.2012.05.019.

[24] P. W. Cleary, M. Sinnott, and R. Morrison. "Prediction of slurry transport in SAG mills using SPH fluid flow in a dynamic DEM based porous media". In: *Minerals Engineering* 19.15 (2006), pp. 1517–1527. DOI: 10.1016/j.mineng.2006.08.018.

[25] R. Clift, J. R. Grace, and M. E. Weber. *Bubbles, drops, and particles*. New York, NY: Academic Press, 1978. ISBN: 978-0-4864-4580-9.

[26] C. Crowe et al. *Multiphase flows with droplets and particles*. 2. ed. Boca Raton: CRC Press, 2011. DOI: https://doi.org/10.1201/b11103.

[27] C. T. Crowe, ed. *Multiphase flow handbook*. Mechanical engineering series. Taylor & Francis, 2006. ISBN: 978-0-8493-1280-9.

[28] C. T. Crowe. "On models for turbulence modulation in fluid–particle flows". In: *International Journal of Multiphase Flow* 26.5 (2000), pp. 719–727. DOI: 10.1016/S0301-9322(99)00050-6.

[29] P. A. Cundall and O. Strack. "A discrete numerical model for granular assemblies". In: *Geotechnique* 29.1 (1979), pp. 47–65.

[30] J. M. Dallavalle. *Micromeritics: The technology of fine particles*. Pitman Publishing Corporation; New York, 1943.

[31] M. Danby, J. Shrimpton, and M. Palmer. "On the optimal numerical time integration for DEM using Hertzian force models". In: *Computers & Chemical Engineering* 58 (2013), pp. 211–222. DOI: 10.1016/j.compchemeng.2013.06.018.

[32] Y. Davit and M. Quintard. “Technical notes on Volume Averaging in Porous Media I: How to Choose a Spatial Averaging Operator for Periodic and Quasiperiodic Structures”. In: *Transport in Porous Media* 119.3 (2017), pp. 555–584. DOI: 10.1007/s11242-017-0899-8.

[33] Y. Davit et al. “Homogenization via formal multiscale asymptotics and volume averaging: How do the two techniques compare?” In: *Advances in Water Resources*. A tribute to Stephen Whitaker 62 (2013), pp. 178–206. DOI: 10.1016/j.advwatres.2013.09.006.

[34] N. G. Deen et al. “Review of discrete particle modeling of fluidized beds”. In: *Chemical Engineering Science* 62.1 (2007), pp. 28–44. DOI: 10.1016/j.ces.2006.08.014.

[35] R. Di Felice. “The voidage function for fluid-particle interaction systems”. In: *International Journal of Multiphase Flow* 20.1 (1994), pp. 153–159. DOI: 10.1016/0301-9322(94)90011-6.

[36] R. Di Felice. “Hydrodynamics of liquid fluidisation”. In: *Chemical Engineering Science* 50.8 (1995), pp. 1213–1245. DOI: 10.1016/0009-2509(95)98838-6.

[37] A. Di Renzo, F. Cello, and F. P. Di Maio. “Simulation of the layer inversion phenomenon in binary liquid–fluidized beds by DEM–CFD with a drag law for polydisperse systems”. In: *Chemical Engineering Science* 66.13 (2011), pp. 2945–2958. DOI: 10.1016/j.ces.2011.03.035.

[38] A. Di Renzo and F. P. Di Maio. “Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes”. In: *Chemical Engineering Science* 59.3 (2004), pp. 525–541. DOI: 10.1016/j.ces.2003.09.037.

[39] A. Di Renzo and F. P. Di Maio. “Homogeneous and bubbling fluidization regimes in DEM–CFD simulations: Hydrodynamic stability of gas and liquid fluidized beds”. In: *Chemical Engineering Science* 62.1 (2007), pp. 116–130. DOI: 10.1016/j.ces.2006.08.009.

[40] C. Drumm et al. “Finite pointset method for simulation of the liquid–liquid flow field in an extractor”. In: *Computers & Chemical Engineering* 32.12 (2008), pp. 2946–2957. DOI: 10.1016/j.compchemeng.2008.03.009.

[41] A. Džiugys and B. Peters. “An approach to simulate the motion of spherical and non-spherical fuel particles in combustion chambers”. In: *Granular Matter* 3.4 (2001), pp. 231–266. DOI: 10.1007/PL00010918.

[42] N. Epstein. “The Voidage Function for the Drag Force Ratio in a Liquid-Fluidized Bed”. In: *The Canadian Journal of Chemical Engineering* 83.3 (2005), pp. 566–572. DOI: 10.1002/cjce.5450830321.

[43] S. Ergun. “Fluid flow through packed columns”. In: *Chem. Eng. Prog.* 48 (1952), pp. 89–94.

[44] “Erratum”. In: *AIChE Journal* 53.11 (2007), pp. 3020–3020. DOI: 10.1002/aic.11330.

[45] L. C. Evans. *Partial differential equations*. Second edition, reprinted with corrections. Graduate studies in mathematics Volume 19. Providence, Rhode Island: American Mathematical Society, 2015. ISBN: 978-0-8218-4974-3.

[46] A. Fabbri et al. “Numerical modeling of heat and mass transfer during coffee roasting process”. In: *Journal of Food Engineering* 105.2 (2011), pp. 264–269. DOI: 10.1016/j.jfoodeng.2011.02.030.

[47] Y. Farjoun and B. Seibold. “An exactly conservative particle method for one dimensional scalar conservation laws”. In: *Journal of Computational Physics* 228.14 (2009), pp. 5298–5315. DOI: 10.1016/j.jcp.2009.04.013.

[48] Y. Q. Feng and A. B. Yu. “Assessment of Model Formulations in the Discrete Particle Simulation of Gas-Solid Flow”. In: *Industrial & Engineering Chemistry Research* 43.26 (2004), pp. 8378–8390. DOI: 10.1021/ie049387v.

[49] J. W. Fernandez et al. “Using SPH one-way coupled to DEM to model wet industrial banana screens”. In: *Minerals Engineering* 24.8 (2011), pp. 741–753. DOI: 10.1016/j.mineng.2011.01.004.

[50] R. O. Fox. “On multiphase turbulence models for collisional fluid–particle flows”. In: *Journal of Fluid Mechanics* 742 (2014), pp. 368–424. DOI: 10.1017/jfm.2014.21.

[51] F. Fraige and P. Langston. “Integration schemes and damping algorithms in distinct element models”. In: *Advanced Powder Technology* 15.2 (2004), pp. 227–245. DOI: 10.1163/156855204773644454.

[52] R. Garg et al. “Open-source MFIX-DEM software for gas–solids flows: Part I—Verification studies”. In: *Powder Technology*. Selected Papers from the 2010 NETL Multiphase Flow Workshop 220 (2012), pp. 122–137. DOI: 10.1016/j.powtec.2011.09.019.

[53] E. Garnier, N. Adams, and P. Sagaut. *Large Eddy Simulation for Compressible Flows*. Dordrecht: Springer Netherlands, 2009. DOI: 10.1007/978-90-481-2819-8.

[54] L. G. Gibilaro et al. “On the apparent viscosity of a fluidized bed”. In: *Chemical Engineering Science*. Fluidized Bed Applications 62.1 (2007), pp. 294–300. DOI: 10.1016/j.ces.2006.08.030.

[55] D. Gidaspow. *Multiphase flow and fluidization: continuum and kinetic theory descriptions*. Boston: Academic Press, 1994. DOI: https://doi.org/10.1016/C2009-0-21244-X.

[56] D. Gidaspow and B. Ettehadieh. “Fluidization in two-dimensional beds with a jet. 2. Hydrodynamic modeling”. In: *Industrial & Engineering Chemistry Fundamentals* 22.2 (1983), pp. 193–201. DOI: 10.1021/i100010a008.

[57] C. Goniva et al. “An Open Source CFD-DEM Perspective”. In: *Proceedings of OpenFOAM Workshop*. Göteborg, 2010, pp. 22–24.

[58] C. Goniva et al. “Influence of rolling friction on single spout fluidized bed simulation”. In: *Particuology* 10.5 (2012), pp. 582–591. DOI: 10.1016/j.partic.2012.05.002.

[59] T. Hales et al. “A formal proof of the Kepler conjecture”. In: *Forum of Mathematics, Pi* 5 (2017). DOI: 10.1017/fmp.2017.1.

[60] J. Harting et al. “Recent advances in the simulation of particle-laden flows”. In: *The European Physical Journal Special Topics* 223.11 (2014), pp. 2253–2267. DOI: 10.1140/epjst/e2014-02262-3.

[61] R. J. Hill, D. L. Koch, and A. J. C. Ladd. “Moderate-Reynolds-number flows in ordered and random arrays of spheres”. In: *Journal of Fluid Mechanics* 448 (2001). DOI: 10.1017/S0022112001005936.

[62] J. E. Hilton, L. R. Mason, and P. W. Cleary. “Dynamics of gas–solid fluidised beds with non-spherical particle geometry”. In: *Chemical Engineering Science* 65.5 (2010), pp. 1584–1596. DOI: 10.1016/j.ces.2009.10.028.

[63] J. E. Hilton and P. W. Cleary. “Comparison of non-cohesive resolved and coarse grain DEM models for gas flow through particle beds”. en. In: *Applied Mathematical Modelling* 38.17 (2014), pp. 4197–4214. DOI: 10.1016/j.apm.2014.02.013.

[64] M. A. v. d. Hoef et al. “Numerical Simulation of Dense Gas-Solid Fluidized Beds: A Multiscale Modeling Strategy”. In: *Annual Review of Fluid Mechanics* 40.1 (2008), pp. 47–70. DOI: 10.1146/annurev.fluid.40.111406.102130.

[65] M. A. van der Hoef, R. Beetstra, and J. A. M. Kuipers. “Lattice-Boltzmann simulations of low-Reynolds-number flow past mono- and bidisperse arrays of spheres: results for the permeability and drag force”. In: *Journal of Fluid Mechanics* 528 (2005), pp. 233–254. DOI: 10.1017/S0022112004003295.

[66] M. A. van der Hoef et al. “Multiscale Modeling of Gas-Fluidized Beds”. In: *Advances in Chemical Engineering*. Vol. 31. Elsevier, 2006, pp. 65–149. DOI: 10.1016/S0065-2377(06)31002-2.

[67] M. Ishii. *Thermo-fluid dynamic theory of two-phase flow*. 1975.

[68] M. Ishii and T. Hibiki. *Thermo-fluid dynamics of two-phase flow*. en. 2. ed. New York: Springer, 2011. ISBN: 978-1-4419-7985-8.

[69] K. Iwashita and M. Oda. “Rolling Resistance at Contacts in Simulation of Shear Band Development by DEM”. In: *Journal of Engineering Mechanics* 124.3 (1998), pp. 285–292. DOI: 10.1061/(ASCE)0733-9399(1998)124:3(285).

[70] K. J. Hanley and C. O’Sullivan. “Analytical study of the accuracy of discrete element simulations: Analytical study of the accuracy of discrete element simulations”. In: *International Journal for Numerical Methods in Engineering* 109.1 (2017), pp. 29–51. DOI: 10.1002/nme.5275.

[71] J. D. Jackson. “Classical Electrodynamics, 3rd ed.” In: *American Journal of Physics* 67.9 (1999), p. 841. DOI: 10.1119/1.19136.

[72] R. Jackson. “Locally averaged equations of motion for a mixture of identical spherical particles and a Newtonian fluid”. In: *Chemical Engineering Science*. Mathematical modelling of chemical and biochemical processes 52.15 (1997), pp. 2457–2469. DOI: 10.1016/S0009-2509(97)00065-1.

[73] A. Jefferies et al. “Finite Pointset Method for the Simulation of a Vehicle Travelling Through a Body of Water”. In: *Meshfree Methods for Partial Differential Equations VII*. Ed. by M. Griebel and M. A. Schweitzer. Springer International Publishing, 2015, pp. 205–221. DOI: 10.1007/978-3-319-06898-5_11.

[74] M. J. Jiang, H. .-.-S. Yu, and D. Harris. “A novel discrete model for granular material incorporating rolling resistance”. In: *Computers and Geotechnics* 32.5 (2005), pp. 340–357. DOI: 10.1016/j.compgeo.2005.05.001.

[75] K. D. Kafui, C. Thornton, and M. J. Adams. “Discrete Particle-Continuum Fluid Modelling for Gas–Solid Fluidised Beds”. In: *Chemical Engineering Science* 57.13 (2002), pp. 2395–2410. DOI: 10.1016/S0009-2509(02)00140-9.

[76] J. B. Keller. “The scope of the image method”. In: *Communications on Pure and Applied Mathematics* 6.4 (1953), pp. 505–512. DOI: 10.1002/cpa.3160060406.

[77] C. Kloss et al. “Models, algorithms and validation for opensource DEM and CFD-DEM”. In: *Progress in Computational Fluid Dynamics, An International Journal* 12.2 (2012), p. 140. DOI: 10.1504/PCFD.2012.047457.

[78] D. L. Koch and R. J. Hill. “Inertial effects in suspensions and porous-media flows”. In: *Annual Review of Fluid Mechanics* 33.1 (2001), pp. 619–647. DOI: 10.1146/annurev.fluid.33.1.619.

[79] H. Kruggel-Emden, F. Stepanek, and A. Munjiza. “Performance of integration schemes in discrete element simulations of particle systems involving consecutive contacts”. In: *Computers & Chemical Engineering* 35.10 (2011), pp. 2152–2157. DOI: 10.1016/j.compchemeng.2010.09.008.

[80] H. Kruggel-Emden, S. Wirtz, and V. Scherer. “A study on tangential force laws applicable to the discrete element method (DEM) for materials with viscoelastic or plastic behavior”. In: *Chemical Engineering Science* 63.6 (2008), pp. 1523–1541. DOI: 10.1016/j.ces.2007.11.025.

[81] H. Kruggel-Emden et al. “Review and extension of normal force models for the Discrete Element Method”. In: *Powder Technology* 171.3 (2007), pp. 157–173. DOI: 10.1016/j.powtec.2006.10.004.

[82] H. Kruggel-Emden et al. “Selection of an appropriate time integration scheme for the discrete element method (DEM)”. In: *Computers & Chemical Engineering* 32.10 (2008), pp. 2263–2279. DOI: 10.1016/j.compchemeng.2007.11.002.

[83] J. Kuhnert. “General Smoothed Particle Hydrodynamics”. PhD thesis. Kaiserslautern University, 1999.

[84] D. Kunii and O. Levenspiel. *Fluidization Engineering*. English. Butterworth-Heinemann, 1991. ISBN: 978-0-08-050664-7.

[85] G. Kuwabara and K. Kono. “Restitution Coefficient in a Collision between Two Spheres”. In: *Japanese Journal of Applied Physics* 26 (Part 1, No. 8 1987), pp. 1230–1233. DOI: 10.1143/JJAP.26.1230.

[86] L. D. Landau et al. *Theory of elasticity*. Pergamon Press, 1986. ISBN: 978-0-08-057069-3.

[87] L. D. Landau, E. M. Lifšic, and L. D. Landau. *Fluid mechanics*. 2. ed., corr. reprint. Course of theoretical physics by L. D. Landau and E. M. Lifshitz ; Vol. 6. Oxford: Pergamon Press, 1989. ISBN: 978-0-08-033932-0.

[88] P. A. Langston, U. Tüzün, and D. M. Heyes. “Continuous potential discrete particle simulations of stress and velocity fields in hoppers: transition from fluid to granular flow”. In: *Chemical Engineering Science* 49.8 (1994), pp. 1259–1275. DOI: 10.1016/0009-2509(94)85095-X.

[89] M. Lemieux et al. “Large-scale numerical investigation of solids mixing in a V-blender using the discrete element method”. In: *Powder Technology*. Particulate Processes in the Pharmaceutical Industry 181.2 (2008), pp. 205–216. DOI: 10.1016/j.powtec.2006.12.009.

[90] L. Li, B. Li, and Z. Liu. “Modeling of spout-fluidized beds and investigation of drag closures using OpenFOAM”. In: *Powder Technology* 305 (2017), pp. 364–376. DOI: 10.1016/j.powtec.2016.10.005.

[91] E. W. C. Lim, C.-H. Wang, and A.-B. Yu. “Discrete element simulation for pneumatic conveying of granular material”. In: *AIChE Journal* 52.2 (2006), pp. 496–509. DOI: 10.1002/aic.10645.

[92] J. M. Link et al. “Flow regimes in a spout–fluid bed: A combined experimental and simulation study”. In: *Chemical Engineering Science* 60.13 (2005), pp. 3425–3442. DOI: 10.1016/j.ces.2005.01.027.

[93] T. Liszka and J. Orkisz. “The finite difference method at arbitrary irregular grids and its application in applied mechanics”. In: *Computers & Structures* 11.1 (1980), pp. 83–95. DOI: 10.1016/0045-7949(80)90149-2.

[94] G. R. Liu and M. B. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific, 2003. DOI: 10.1142/5340.

[95] E. Loth. "Numerical approaches for motion of dispersed particles, droplets and bubbles". In: *Progress in Energy and Combustion Science* 26.3 (2000), pp. 161–223. DOI: 10.1016/S0360-1285(99)00013-1.

[96] G. Lu, J. Third, and C. Müller. "Discrete element models for non-spherical particle systems: From theoretical developments to applications". In: *Chemical Engineering Science* 127 (2015), pp. 425–465. DOI: 10.1016/j.ces.2014.11.050.

[97] S. Luding. "Introduction to discrete element methods: basic of contact force models and how to perform the micro-macro transition to continuum theory". In: *European Journal of Environmental and Civil Engineering* 12.7 (2008), pp. 785–826. DOI: 10.1080/19648189.2008.9693050.

[98] I. Macdonald et al. "Flow through porous media-the Ergun equation revisited". In: *Industrial & Engineering Chemistry Fundamentals* 18.3 (1979), pp. 199–208.

[99] K. F. Malone and B. H. Xu. "Determination of contact parameters for discrete element method simulations of granular systems". In: *Particuology*. Simulation and Modeling of Particulate Systems 6.6 (2008), pp. 521–528. DOI: 10.1016/j.partic.2008.07.012.

[100] D. Markauskas et al. "Comparative study on mesh-based and mesh-less coupled CFD-DEM methods to model particle-laden flow". In: *Powder Technology* 305 (2017), pp. 78–88. DOI: 10.1016/j.powtec.2016.09.052.

[101] C. Marle. "Ecoulements monophasiques en milieu poreux". In: *Rev. Inst. Francais du Petrole* 22.10 (1967), pp. 1471–1509.

[102] F. Mashayek and R. Pandya. "Analytical description of particle/droplet-laden turbulent flows". In: *Progress in Energy and Combustion Science* 29.4 (2003), pp. 329–378. DOI: 10.1016/S0360-1285(03)00029-7.

[103] H.-G. Matuttis and J. Chen. *Understanding the Discrete Element Method: Simulation of Non-Spherical Particles for Granular and Multi-Body Systems*. Singapore: John Wiley & Sons (Asia) Pte. Ltd, 2014. ISBN: 978-1-118-56721-0.

[104] L. Mazzei. "Recent Advances in Modeling Gas-Particle Flows". en. In: *Handbook of Multiphase Flow Science and Technology*. Singapore: Springer Singapore, 2016, pp. 1–43. DOI: 10.1007/978-981-4585-86-6_8-1.

[105] I. Michel et al. "Meshfree generalized finite difference methods in soil mechanics—part II: numerical results". In: *GEM - International Journal on Geomathematics* 8.2 (2017), pp. 191–217. DOI: 10.1007/s13137-017-0096-5.

[106] B. K. Mishra and R. K. Rajamani. "The discrete element method for the simulation of ball mills". In: *Applied Mathematical Modelling* 16.11 (1992), pp. 598–604. DOI: 10.1016/0307-904X(92)90035-2.

[107] P. A. Moysey and M. R. Thompson. "Modelling the solids inflow and solids conveying of single-screw extruders using the discrete element method". In: *Powder Technology* 153.2 (2005), pp. 95–107. DOI: 10.1016/j.powtec.2005.03.001.

[108] H. R. Norouzi et al. *Coupled CFD-DEM modeling: formulation, implementation and application to multiphase flows*. First edition. Chichester, West Sussex: Wiley, 2016. DOI: 10.1002/9781119005315.

[109] C. O'Sullivan. *Particulate discrete element modelling: a geomechanics perspective*. eng. Applied geotechnics 4. London: Spon Press, 2011. DOI: https://doi.org/10.1201/9781482266498.

[110] C. O'Sullivan and J. D. Bray. "Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme". In: *Engineering Computations* (2004). DOI: 10.1108/02644400410519794.

[111] I. Ostermann et al. "Meshfree generalized finite difference methods in soil mechanics—part I: theory". In: *GEM - International Journal on Geomathematics* 4.2 (2013), pp. 167–184. DOI: 10.1007/s13137-013-0048-7.

[112] M. Otsubo, C. O'Sullivan, and T. Shire. "Empirical assessment of the critical time increment in explicit particulate discrete element method simulations". In: *Computers and Geotechnics* 86 (2017), pp. 67–79. DOI: 10.1016/j.compgeo.2016.12.022.

[113] P. J. Owen and P. W. Cleary. "Prediction of screw conveyor performance using the Discrete Element Method (DEM)". In: *Powder Technology*. Special Issue: Discrete Element Methods: The 4th International conference on Discrete Element Methods 193.3 (2009), pp. 274–288. DOI: 10.1016/j.powtec.2009.03.012.

[114] N. A. Patankar and D. D. Joseph. "Modeling and numerical simulation of particulate flows by the Eulerian–Lagrangian approach". In: *International Journal of Multiphase Flow* 27.10 (2001), pp. 1659–1684. DOI: 10.1016/S0301-9322(01)00021-0.

[115] Z. Peng et al. "Influence of void fraction calculation on fidelity of CFD-DEM simulation of gas-solid bubbling fluidized beds". In: *AIChE Journal* 60.6 (2014), pp. 2000–2018. DOI: 10.1002/aic.14421.

[116] V. L. Popov. *Contact mechanics and friction: physical principles and applications*. en. Berlin: Springer, 2010. DOI: https://doi.org/10.1007/978-3-642-10803-7.

[117] T. Pöschel and T. Schwager. *Computational granular dynamics: models and algorithms*. eng. Berlin: Springer, 2005. DOI: 10.1007/3-540-27720-X.

[118] A. Prosperetti and G. Tryggvason, eds. *Computational Methods for Multiphase Flow*. Cambridge: Cambridge University Press, 2007. DOI: 10.1017/CBO9780511607486.

[119] M. Quintard and S. Whitaker. "Transport in ordered and disordered porous media I: The cellular average and the use of weighting functions". In: *Transport in Porous Media* 14.2 (1994), pp. 163–177. DOI: 10.1007/BF00615199.

[120] M. Quintard and S. Whitaker. "Transport in ordered and disordered porous media II: Generalized volume averaging". In: *Transport in Porous Media* 14.2 (1994), pp. 179–206. DOI: 10.1007/BF00615200.

[121] S. Radl et al. "State of the Art in Mapping Schemes for Dilute and Dense Euler-Lagrange Simulations". English. In: *Progress in Applied CFD*. SINTEF Academic Press, 2015, pp. 103–112.

[122] E. O. Reséndiz-Flores, J. Kuhnert, and F. R. Saucedo-Zendejo. "Application of a generalized finite difference method to mould filling process". In: *European Journal of Applied Mathematics* 29.3 (2018), pp. 450–469. DOI: 10.1017/S0956792517000249.

[123] R. E. Roberson and R. Schwertassek. *Dynamics of Multibody Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988. DOI: 10.1007/978-3-642-86464-3.

[124] M. Robinson, M. Ramaioli, and S. Luding. "Fluid–particle flow simulations using two-way-coupled mesoscale SPH–DEM and validation". In: *International Journal of Multiphase Flow* 59 (2014), pp. 121–134. DOI: 10.1016/j.ijmultiphaseflow.2013.11.003.

[125] L. Rong, K. Dong, and A. Yu. “Lattice-Boltzmann simulation of fluid flow through packed beds of uniform spheres: Effect of porosity”. In: *Chemical Engineering Science* 99 (2013), pp. 44–58. DOI: 10.1016/j.ces.2013.05.036.

[126] E. Rougier, A. Munjiza, and N. W. M. John. “Numerical comparison of some explicit time integration schemes used in DEM, FEM/DEM and molecular dynamics”. In: *International Journal for Numerical Methods in Engineering* 61.6 (2004), pp. 856–879. DOI: 10.1002/nme.1092.

[127] M. Saidi et al. “Hydrodynamic investigation of gas-solid flow in rectangular spout-fluid bed using CFD-DEM modeling”. In: *Powder Technology* 284 (2015), pp. 355–364. DOI: 10.1016/j.powtec.2015.07.005.

[128] M. Sakai and S. Koshizuka. “Large-scale discrete element modeling in pneumatic conveying”. In: *Chemical Engineering Science* 64.3 (2009), pp. 533–539. DOI: 10.1016/j.ces.2008.10.003.

[129] M. Sakai et al. “Verification and validation of a coarse grain model of the DEM in a bubbling fluidized bed”. In: *Chemical Engineering Journal* 244 (2014), pp. 33–43. DOI: 10.1016/j.cej.2014.01.029.

[130] S. Sarkar, M. van der Hoef, and J. Kuipers. “Fluid–particle interaction from lattice Boltzmann simulations for flow through polydisperse random arrays of spheres”. In: *Chemical Engineering Science* 64.11 (2009), pp. 2683–2691. DOI: 10.1016/j.ces.2009.02.045.

[131] F. R. Saucedo-Zendejo and E. O. Reséndiz-Flores. “A new approach for the numerical simulation of free surface incompressible flows using a meshfree method”. In: *Computer Methods in Applied Mechanics and Engineering* 324 (2017), pp. 619–639. DOI: 10.1016/j.cma.2017.06.027.

[132] J. Schäfer, S. Dippel, and D. E. Wolf. “Force Schemes in Simulations of Granular Materials”. In: *Journal de Physique I* 6.1 (1996), pp. 5–20. DOI: 10.1051/jp1:1996129.

[133] L. Schiller and A. Z. Naumann. “Über die grundlegenden Berechnungen bei der Schwerkraftaufbereitung”. In: *Ver. Deut. Ing.* 77 (1933), pp. 318–320.

[134] T. Schwager and T. Pöschel. “Coefficient of restitution and linear–dashpot model revisited”. In: *Granular Matter* 9.6 (2007), pp. 465–469. DOI: 10.1007/s10035-007-0065-z.

[135] B. Seibold. “M-Matrices in Meshless Finite Difference Methods”. PhD thesis. Kaiserslautern University, 2006.

[136] T. Seifarth. *Numerische Algorithmen für gitterfreie Methoden zur Lösung von Transportproblemen*. Stuttgart: Fraunhofer Verlag, 2018. ISBN: 978-3-8396-1269-9.

[137] M. Sinnott, P. W. Cleary, and R. D. Morrison. “Slurry flow in a tower mill”. In: *Minerals Engineering* 24.2 (2011), pp. 152–159. DOI: 10.1016/j.mineng.2010.11.002.

[138] J. C. Slattery. “Flow of viscoelastic fluids through porous media”. In: *AIChE Journal* 13.6 (1967), pp. 1066–1071. DOI: 10.1002/aic.690130606.

[139] V. Šmilauer and B. Chareyre. “DEM formulation”. In: *Yade Documentation 2nd ed*. The Yade Project, 2015. DOI: 10.5281/zenodo.34044.

[140] Sommerfeld Arnold. *Partial Differential Equations In Physics*. Academic Press Inc. Publishers New York, 1949.

[141] J. Spurk and N. Aksel. *Fluid Mechanics*. Springer-Verlag Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-73537-3.

[142] J. Spurk and N. Aksel. *Strömungslehre*. Springer-Lehrbuch. Springer-Verlag Berlin Heidelberg, 2010. DOI: 10.1007/978-3-642-13143-1.

[143] A. B. Subramanyam et al. "On parallelization and load balancing aspects of the finite-pointset method". In: *International Journal of Computer Mathematics* 88.2 (2011), pp. 360–374. DOI: 10.1080/00207160903452210.

[144] P. Suchde. *Conservation and accuracy in meshfree generalized finite difference methods*. In collab. with F.-I. für Techno- und Wirtschaftsmathematik. Stuttgart: Fraunhofer Verlag, 2018. ISBN: 978-3-8396-1325-2.

[145] P. Suchde and J. Kuhnert. "Point Cloud Movement For Fully Lagrangian Meshfree Methods". In: *Journal of Computational and Applied Mathematics* 340 (2018), pp. 89–100. DOI: 10.1016/j.cam.2018.02.020.

[146] P. Suchde, J. Kuhnert, and S. Tiwari. "On meshfree GFDM solvers for the incompressible Navier–Stokes equations". In: *Computers & Fluids* 165 (2018), pp. 1–12. DOI: 10.1016/j.compfluid.2018.01.008.

[147] P. Suchde et al. "A flux conserving meshfree method for conservation laws". In: *International Journal for Numerical Methods in Engineering* 112.3 (2017), pp. 238–256. DOI: 10.1002/nme.5511.

[148] R. Sun and H. Xiao. "Diffusion-based coarse graining in hybrid continuum–discrete solvers: Applications in CFD–DEM". In: *International Journal of Multiphase Flow* 72 (2015), pp. 233–247. DOI: 10.1016/j.ijmultiphaseflow.2015.02.014.

[149] R. Sun and H. Xiao. "Diffusion-based coarse graining in hybrid continuum–discrete solvers: Theoretical formulation and a priori tests". In: *International Journal of Multiphase Flow* 77 (2015), pp. 142–157. DOI: 10.1016/j.ijmultiphaseflow.2015.08.014.

[150] R. Sun and H. Xiao. "SediFoam: A general-purpose, open-source CFD–DEM solver for particle-laden flow with emphasis on sediment transport". In: *Computers & Geosciences* 89 (2016), pp. 207–219. DOI: 10.1016/j.cageo.2016.01.011.

[151] X. Sun, M. Sakai, and Y. Yamada. "Three-dimensional simulation of a solid–liquid flow by the DEM–SPH method". In: *Journal of Computational Physics* 248 (2013), pp. 147–176. DOI: 10.1016/j.jcp.2013.04.019.

[152] C. Thornton, S. J. Cummins, and P. W. Cleary. "An investigation of the comparative behaviour of alternative contact force models during inelastic collisions". In: *Powder Technology* 233 (2013), pp. 30–46. DOI: 10.1016/j.powtec.2012.08.012.

[153] S. Tiwari and J. Kuhnert. "Finite Pointset Method Based on the Projection Method for Simulations of the Incompressible Navier-Stokes Equations". In: *Meshfree Methods for Partial Differential Equations*. Vol. 26. Lecture notes in Computational Science and Engineering. Springer Berlin Heidelberg, 2003, pp. 373–387. DOI: 10.1007/978-3-642-56103-0_26.

[154] S. Torquato, T. M. Truskett, and P. G. Debenedetti. "Is Random Close Packing of Spheres Well Defined?" In: *Physical Review Letters* 84.10 (2000), pp. 2064–2067. DOI: 10.1103/PhysRevLett.84.2064.

[155] A. Tramecon and J. Kuhnert. “Simulation of Advanced Folded Airbags with VPS-PAM-CRASH/FPM: Development and Validation of Turbulent Flow Numerical Simulation Techniques Applied to Curtain Bag Deployments”. In: SAE 2013 World Congress & Exhibition. 2013, pp. 2013–01–1158. DOI: 10.4271/2013-01-1158.

[156] Y. Tsuji, T. Kawaguchi, and T. Tanaka. “Discrete particle simulation of two-dimensional fluidized bed”. In: *Powder Technology* 77.1 (1993), pp. 79–87. DOI: 10.1016/0032-5910(93)85010-7.

[157] Y. Tsuji, T. Tanaka, and T. Ishida. “Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe”. In: *Powder Technology* 71.3 (1992), pp. 239–250. DOI: 10.1016/0032-5910(92)88030-L.

[158] R. Tuley et al. “On the optimal numerical time integration for Lagrangian DEM within implicit flow solvers”. In: *Computers & Chemical Engineering* 34.6 (2010), pp. 886–899. DOI: 10.1016/j.compchemeng.2009.10.003.

[159] E. Uhlmann, R. Gerstenberger, and J. Kuhnert. “Cutting Simulation with the Meshfree Finite Pointset Method”. In: *Procedia CIRP* 8 (2013), pp. 391–396. DOI: 10.1016/j.procir.2013.06.122.

[160] K. Vollmari et al. “Pressure drop investigations in packings of arbitrary shaped particles”. In: *Powder Technology* 271 (2015), pp. 109–124. DOI: 10.1016/j.powtec.2014.11.001.

[161] B. G. M. van Wachem and A. E. Almstedt. “Methods for multiphase computational fluid dynamics”. In: *Chemical Engineering Journal*. Festschrift Prof. Cor M. van den Bleek 96.1 (2003), pp. 81–98. DOI: 10.1016/j.cej.2003.08.025.

[162] B. G. M. van Wachem et al. “Experimental validation of Lagrangian–Eulerian simulations of fluidized beds”. In: *Powder Technology* 116.2 (2001), pp. 155–165. DOI: https://doi.org/10.1016/S0032-5910(00)00389-2.

[163] C. Y. Wen and Y. H. Yu. “Mechanics of fluidization”. In: *Chem. Eng. Prog. Symp. Ser.* Vol. 62. 1966, pp. 100–111.

[164] S. Whitaker. “Diffusion and dispersion in porous media”. In: *AIChE Journal* 13.3 (1967), pp. 420–427. DOI: 10.1002/aic.690130308.

[165] S. Whitaker. *The Method of Volume Averaging*. Red. by J. Bear. Vol. 13. Theory and Applications of Transport in Porous Media. Dordrecht: Springer Netherlands, 1999. DOI: 10.1007/978-94-017-3389-2.

[166] S. Whitaker. “Advances in theory of fluid motion in porous media”. In: *Industrial & Engineering Chemistry* 61.12 (1969), pp. 14–28. DOI: 10.1021/ie50720a004.

[167] J. Wittenburg. *Dynamics of Multibody Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. ISBN: 978-3-540-73913-5.

[168] H. Xiao and J. Sun. “Algorithms in a Robust Hybrid CFD-DEM Solver for Particle-Laden Flows”. In: *Communications in Computational Physics* 9.2 (2011), pp. 297–323. DOI: 10.4208/cicp.260509.230210a.

[169] K. Yazdchi and S. Luding. “Towards unified drag laws for inertial flow through fibrous materials”. In: *Chemical Engineering Journal* 207-208 (2012), pp. 35–48. DOI: 10.1016/j.cej.2012.06.140.

[170] D. Z. Zhang and A. Prosperetti. “Averaged equations for inviscid disperse two-phase flow”. In: *Journal of Fluid Mechanics* 267.-1 (1994), p. 185. DOI: 10.1017/S0022112094001151.

[171] D. Z. Zhang and A. Prosperetti. “Momentum and energy equations for disperse two-phase flows and their closure for dilute suspensions”. In: *International Journal of Multiphase Flow* 23.3 (1997), pp. 425–453. DOI: 10.1016/S0301-9322(96)00080-8.

[172] J. Zhao and T. Shan. “Coupled CFD–DEM simulation of fluid–particle interaction in geomechanics”. In: *Powder Technology* 239 (2013), pp. 248–258. DOI: 10.1016/j.powtec.2013.02.003.

[173] W. Zhong et al. “CFD simulation of dense particulate reaction system: Approaches, recent advances and applications”. In: *Chemical Engineering Science* 140 (2016), pp. 16–43. DOI: 10.1016/j.ces.2015.09.035.

[174] W. Zhong et al. “DEM/CFD-DEM Modelling of Non-spherical Particulate Systems: Theoretical Developments and Applications”. In: *Powder Technology* 302 (2016), pp. 108–152. DOI: 10.1016/j.powtec.2016.07.010.

[175] Z. Y. Zhou et al. “Discrete particle simulation of particle–fluid flow: model formulations and their applicability”. In: *Journal of Fluid Mechanics* 661 (2010), pp. 482–510. DOI: 10.1017/S002211201000306X.

[176] H. P. Zhu and A. B. Yu. “Averaging method of granular materials”. In: *Physical Review E* 66.2 (2002). DOI: 10.1103/PhysRevE.66.021302.

[177] H. P. Zhu et al. “Discrete particle simulation of particulate systems: Theoretical developments”. In: *Chemical Engineering Science*. Frontier of Chemical Engineering - Multi-scale Bridge between Reductionism and Holism 62.13 (2007), pp. 3378–3396. DOI: 10.1016/j.ces.2006.12.089.

Academic CV

04/2013 - 12/2015	**Master of Science in Industrial Mathematics** University of Kaiserslautern
09/2013 - 12/2013	**Semester abroad** Polytechnic Institute of New York University New York City
04/2010 - 03/2013	**Bachelor of Science in Mathematics** University of Kaiserslautern
03/2010	**Abitur** (general qualification for university entrance) Herzog-Johann-Gymnasium Simmern

Wissenschaftlicher Werdegang

04/2013 - 12/2015	**Master of Science in Technomathematik** Technische Universität Kaiserslautern
09/2013 - 12/2013	**Auslandssemester** Polytechnic Institute of New York University New York City
04/2010 - 03/2013	**Bachelor of Science in Mathematik** Technische Universität Kaiserslautern
03/2010	**Abitur** Herzog-Johann-Gymnasium Simmern

We consider the modeling and simulation of flows composed of a fluid with an immersed particulate solid phase within a two-way coupled scheme. We embed it into the generalized finite difference framework of the finite pointset method (FPM). Both phases are described in a Lagrangian formalism and are represented by point clouds. This allows us to treat all phases in a common framework and to take advantage of synergies in terms of data structures and algorithms. A key challenge which is introduced by the generalized finite difference setting is the calculation of averaged quantities. Due to the properties of our mesh-free approach which is missing an inherent definition of cell volume, conventional averaging strategies from mesh-based schemes are not directly applicable. We employ an approach which circumvents these problems and takes the finite difference nature of the FPM into account. Additionally, we bring to light the required changes to a projection method for the fluid phase to incorporate the multi-phase setting. The solid phase solver, averaging scheme, and fluid solver are embedded into a coupled algorithm with a substepping procedure to improve efficiency.

ISBN 978-3-8396-1674-1

9 783839 616741

FRAUNHOFER VERLAG