Modeling and Simulation of Suspension Dynamics Capturing Shapes and Contacts

With Applications

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für Chemieingenieurwesen und Verfahrenstechnik des
Karlsruher Instituts für Technologie (KIT)
genehmigte

Dissertation

von
M. Sc. Jan Eric Marquardt
aus Paderborn

Tag der mündlichen Prüfung: 19. März 2025

Erstgutachter:	PD Dr. rer. nat. Mathias J. Krause
Zweitgutachter:	Prof. Dr. Markus Uhlmann

Zusammenfassung

Partikelströmungen sind für viele natürliche Prozesse von zentraler Bedeutung, von Nanopartikeln in der Luft bis hin zu großräumigen Sedimentablagerungen in den Ozeanen. Ihre Bedeutung erstreckt sich auch auf viele industrielle Anwendungen, z.B. bei der Verarbeitung von Lebensmitteln, Mineralien und Chemikalien, in der Biotechnologie, bei Energieanwendungen und bei der Herstellung von Farben, Kunststoffen und Kosmetika. Ein tieferes Verständnis der komplexen Dynamik dieser Systeme ist daher wichtig und bietet erhebliche Vorteile für die Umwelt und technologische Fortschritte.

Diese Arbeit befasst sich mit nicht-kolloidale Suspensionen, bei denen Feststoffpartikel mit vernachlässigbarer Brownscher Bewegung in einer Flüssigkeit dispergiert sind. Diese Systeme sind in industriellen Prozessen und technischen Anwendungen weit verbreitet, wo sie eine Schlüsselrolle bei der Gestaltung von Materialeigenschaften, Prozesseffizienz und Produktqualität einnehmen. In vielen dieser Anwendungen haben die Partikel jedoch komplexe Formen, die die Dynamik der Suspensionen erheblich beeinflussen. Die derzeitige Forschung zur Dynamik von Suspensionen befasst sich jedoch größtenteils nur mit vereinfachten, meist kugelförmigen, Geometrien. Wenn komplexe Partikelformen betrachtet werden, dann oft nur in geringer Anzahl oder die Untersuchungen konzentrieren sich auf qualitative Beschreibungen und nicht auf quantitative Bewertungen.

Das Hauptziel dieser Arbeit ist daher die Entwicklung und Validierung eines neuartigen Modells, das die Dynamik von Suspensionen mit hohen Partikelvolumenanteilen (bis zu 30%) erfassen kann und gleichzeitig beliebige konvexe Partikelformen zulässt. Um dieses Ziel zu erreichen, werden in dieser Arbeit fünf Schlüsselbeiträge vorgestellt, die auf vier Einzelarbeiten basieren.

Der erste Beitrag befasst sich mit der Herausforderung der Leistung bei der

Simulation einer großen Anzahl von vollständig aufgelösten Partikeln. In dieser Arbeit wird die Lattice-Boltzmann-Methode (LBM) verwendet, die für ihre hohe Effizienz bekannt ist. Konkret wird für Partikelströmungen die LBM-basierte homogenisierten Lattice-Boltzmann-Methode (HLBM) verwendet. Obwohl HLBM effektiv ist, nutzt es die Möglichkeiten von LBM für Hochleistungssimulationen noch nicht vollständig aus. Daher wird ein mit HLBM kompatibles Schema für die Zerlegung oberflächenaufgelöster Partikel eingeführt. Dieses Schema reduziert den Kommunikations- und Rechenaufwand für partikelbezogene Operationen erheblich erheblich, indem es die Kommunikation auf eine definierte Nachbarschaft jeder Verarbeitungseinheit beschränkt, anstatt alle Einheiten miteinander kommunizieren zu lassen. Dieser Ansatz führt zu bis zu fünfmal schnelleren Simulationen und ermöglicht die Simulation einer wesentlich größeren Anzahl von oberflächenaufgelösten Partikeln.

Der zweite Beitrag stellt ein neuartiges diskretes Kontaktmodell vor, welches für komplexe konvexe Partikelformen entwickelt wurde und mit HLBM kompatibel ist. Durch die Integration des nichtlinearen Kontaktmodells wird die Beschränkung von HLBM auf eine Zweiwegekopplung und damit auf moderate Partikelvolumenanteile überwunden, so dass vollständig aufgelöste Partikelströmungssimulationen mit Vierwegekopplung möglich werden. Das Modell berücksichtigt sowohl die elastische Verformung als auch die viskose Dämpfung der interagierenden Festkörper und ermittelt sowohl Normal- als auch Tangentialkräfte durch Diskretisierung des Kontaktbereichs. Die Validierung anhand von Trocken- und Nasskontakten bestätigt die Robustheit, Vielseitigkeit und Genauigkeit der entwickelten Kontaktbehandlung.

Der dritte Beitrag ist die Realisierung eines einheitlichen modularen Partikel-Frameworks für die Simulation von Partikelströmungen mit Vierwegekopplung. Implementiert in der Open-Source-Software OpenLB bietet dieses Framework eine flexible Architektur, die in der Lage ist, eine Vielzahl von Partikelströmungsproblemen effizient zu lösen. Das modulare Design ermöglicht eine einfache Erweiterbarkeit und Anpassung an unterschiedliche Simulationsanforderungen, einschließlich vollständig aufgelöster und subskaliger Partikelsimulationen, obwohl sich diese Arbeit hauptsächlich auf erstere konzentriert. Darüber hinaus sorgt die Integration automatisierter Tests für kontinuierliche Qualität und Zuverlässigkeit, so dass die Software robust und an künftige Entwicklungen und veränderte Forschungsanforderungen anpassbar bleibt.

Der vierte Beitrag demonstriert die Anwendung des entwickelten direkten numerischen Simulationsansatzes auf die Untersuchung der Schwarmsedimentation. Die Untersuchung vergleicht das Sedimentationsverhalten von Kugel- und Würfelschwärmen für Archimedes-Zahlen zwischen 500 und 2000 und Partikelvolumenfraktionen zwischen 10% und 30%. Die Ergebnisse zeigen, dass sich Würfel um 13% bis 26% langsamer sedimentieren als Schwärme volumengleicher Kugeln, während Kugeln eine stärkere Tendenz zur Clusterbildung aufweisen. Diese Ergebnisse unterstreichen die starke Abhängigkeit des Sedimentationsverhaltens von der Partikelform und deuten darauf hin, dass die Form eine entscheidende Rolle bei Trennprozessen spielt.

Der abschließende Beitrag gibt einen umfassenden Überblick über den entwickelten Ansatz und betrachtet verschiedene seiner berichteten Anwendungen, um Stärken und Schwächen zu identifizieren. Zu den Stärken zählen der monolithische Charakter des Ansatzes, der eine nahtlose Simulation verschiedener Szenarien mit Fluid-Festkörper-Wechselwirkungen ermöglicht und damit den Modellierungsprozess effizient gestaltet, sowie die Fähigkeit, beliebige konvexe Partikelformen in Simulationen mit Vierwegekopplung und beliebige Formen in Simulationen mit Zweiwegekopplung zu handhaben und dabei lokale Partikel-Fluid-Wechselwirkungen an der Partikeloberfläche detailliert abzubilden. Das parallele Design gewährleistet zudem eine hervorragende Recheneffizienz und eignet sich daher für umfangreiche Simulationen. Allerdings führt die Notwendigkeit kleiner Zeitschritte aufgrund der diffusiven Skalierung zu einem relativ hohen Rechenaufwand. Darüber hinaus begrenzt die derzeitige Beschränkung auf starre Partikel die Anwendbarkeit in Szenarien mit deformierbaren oder flexiblen Objekten.

Diese Beiträge führen zu einem robusten, vielseitigen und gut dokumentierten Framework für die Simulation von Partikelströmungen mit komplexen Partikelformen. Dieses Framework wurde bereits in mehreren ingenieurwissenschaftlichen Anwendungsstudien eingesetzt und hat wertvolle Erkenntnisse zur Verbesserung der Prozesseffizienz und Produktqualität geliefert. Diese Studien zeigen auch die Fähigkeit des Frameworks, eine Vielzahl von Szenarien zu behandeln, die sowohl Suspensionen als auch Aerosole beinhalten. Die Vielseitigkeit und die freie Verfügbarkeit des Frameworks machen es zu einem wertvollen Werkzeug für die zukünftige industrielle und akademische Forschung.

Abstract

Particulate flows are central to many natural processes, ranging from nanoparticles in the air to large areas of sedimentation in the oceans. Their importance also extends to numerous industrial applications, including food, mineral and chemical processing, biotechnology, energy applications, and the manufacture of paints, plastics, and cosmetics. A deeper understanding of the complex dynamics of these systems is therefore important and offers significant environmental benefits and technological advances.

This thesis focuses on non-colloidal suspensions, where solid particles with negligible Brownian motion are dispersed in a liquid. These systems are ubiquitous in industrial processes and engineering applications where they play a key role in shaping material properties, process efficiency and product quality. In many of these applications, the particles have complex shapes that significantly influence the dynamics of the suspensions. However, current research on suspension dynamics is mostly only concerned with simplified, usually spherical, geometries. When complex particle shapes are considered, it is usually only in small numbers, or the studies focus on qualitative descriptions rather than quantitative evaluations.

The primary objective of this work is therefore to develop and validate a novel model capable of capturing suspension dynamics with high particle volume fractions (up to 30%), while allowing for arbitrary convex particle shapes. To achieve this goal, the thesis presents five key contributions, which are based on four individual works.

The first contribution addresses the performance challenge when simulating large numbers of surface resolved particles. This work uses the lattice Boltzmann method (LBM), which is known for its high efficiency. Specifically, for particulate flows, the LBM-based homogenized lattice Boltzmann method

(HLBM) is used. However, while HLBM is effective, it has yet to take full advantage of the capabilities of LBM for high-performance simulations. Therefore, a surface resolved particle decomposition scheme compatible with HLBM is introduced. This scheme significantly reduces the communication and computational overhead of particle-related operations by restricting communication to a defined neighborhood of each processing unit, rather than having all units communicate with each other. The approach results in simulations that are up to five times faster, making it possible to simulate a significantly larger number of surface resolved particles.

The second contribution introduces a novel discrete contact model designed for complex convex particle shapes and compatible with HLBM. By integrating the nonlinear contact model, the limitation of HLBM to two-way coupling and thus moderate particle volume fractions is overcome, allowing for four-way coupled, fully resolved particulate flow simulations. The model accounts for elastic deformation and viscous damping aspects of interacting solids, and evaluates both normal and tangential forces by discretizing the contact region. The validation with dry and wet contacts confirms the robustness, the versatility, and the accuracy of the developed contact treatment.

The third contribution is the realization of a unified modular particle framework for four-way coupled particulate flow simulations. Implemented in the open source software OpenLB, this framework provides a flexible architecture capable of efficiently solving various particulate flow problems. The modular design allows for easy extensibility and adaptation to different simulation needs, including both fully resolved and subgrid scale particle simulations, although this work primarily focuses on the former. In addition, the integration of automated testing ensures ongoing quality and reliability, keeping the software robust and adaptable to future developments and evolving research requirements.

The fourth contribution demonstrates the application of the developed direct numerical simulation approach to the study of hindered settling. The investigation compares the settling behavior of swarms of spheres and cubes for Archimedes numbers between 500 and 2000 and particle volume fractions between 10% and 30%. The results show that cubes settle 13% to 26% slower than swarms of volume-equivalent spheres, while spheres have a higher tendency to cluster. These results underscore the strong dependence of settling behavior on particle shape and suggest that shape plays a critical role in separation processes.

The final contribution provides a comprehensive review of the developed

approach and an examination of various reported applications to identify strengths and limitations. Key strengths include the monolithic nature of the approach, which allows seamless simulation of diverse fluid-solid interaction scenarios, streamlining the modeling process, and its ability to handle arbitrary convex particle shapes in four-way coupled simulations and arbitrary shapes in two-way coupled simulations while accurately representing local surface interactions. The parallel design also ensures superior computational efficiency, making it suitable for large-scale simulations. However, the requirement for small time steps due to diffusive scaling results in relatively high computational effort. Furthermore, the current limitation to rigid particles restricts its applicability in scenarios involving deformable or flexible entities.

These contributions result in a robust, versatile, and well-documented framework for simulating particulate flows with complex particle shapes. This framework has already been used in several studies of engineering applications, providing valuable insights that support improvements in process efficiency and product quality. These studies also demonstrate the framework's ability to handle a wide variety of scenarios involving both suspensions and aerosols. The versatility and free availability of the framework make it a valuable tool for future industrial and academic research.

Contents

1
Introduction

Particle systems are an essential part of a wide variety of natural processes, ranging from tiny nanoparticles in the air [1–6] to vast areas of sedimentation in the oceans [7, 8]. They are also central to a wide range of technological applications in various sectors of the process industries, including food, mineral and chemical processing, biotechnology, energy applications, and the manufacture of paints, plastics, and cosmetics [9]. In the pharmaceutical sector, for example, the control and understanding of particulate flows play a key role in drug formulation [10]. Understanding the complex dynamics of these systems is therefore not just an academic endeavor, but a practical necessity with far-reaching potential benefits, from environmental protection to technological innovation.

1.1 Suspension Dynamics

Recognizing the inherent complexity of particle systems, the focus of this work is on the specific subset of non-colloidal suspensions, which are defined in Section 1.1.1 along with related terminology. The focus on non-colloidal suspensions is motivated by their significant relevance in various fields, such as mechanical process engineering, as discussed in Section 1.1.2, and the open research questions, especially when dealing with particles of complex shapes, as identified in the review of the current state of the art in Section 1.1.3.

1.1.1 Definitions

Dispersions are multiphase systems in which one or more of these phases are separated into smaller domains and distributed throughout a continuous phase [11]. Within the broad category of dispersions, there are several subtypes that are classified based on the physical state of the dispersed and continuous phases. One such subtype, particularly relevant to this work, is a suspension in which solid particles are dispersed in a liquid [11]. In this context, it is more common to use the term component rather than phase to refer to the different parts of the system.

Suspensions can be further categorized based on the behavior of the dispersed particles. Colloidal suspensions, often also referred to as colloids, contain particles small enough to be significantly affected by Brownian motion. Typically, colloidal particles have diameters in the approximate range of 1 nm to 1 µm [12]. Conversely, non-colloidal suspensions refer to systems in which the Brownian motion of the solid particles is negligible and generally involve particles with diameters exceeding 1 µm. It is important to note that these size ranges are approximations and depend on system parameters such as viscosity and temperature [13].

Due to the scope of this thesis, the term “suspension” will be used in the following to refer specifically to non-colloidal suspensions unless otherwise noted.

1.1.2 Relevance

Non-colloidal suspensions are ubiquitous in numerous industrial processes and engineering applications, playing a pivotal role in shaping material properties, process efficiency, and product quality. From the construction industry’s reliance on concrete and mortar formulations containing suspended aggregates [14] to the intricate operations of mineral processing where coarse particles undergo grinding and flotation [15], the presence of non-colloidal suspensions permeates diverse fields. In process engineering, the behavior of suspensions governs the effectiveness of separation techniques such as filtration and centrifugation [16]. In addition, non-colloidal suspensions are common in food processing [17] where they have a major impact on texture and stability [18], thus playing an important role in the overall quality of the final product. These applications, among others, often involve complex non-spherical particle shapes, which further influence the dynamics and behavior of the suspensions [19].

1.1.3 State of the Art

Due to their importance in various industrial processes and engineering applications, non-colloidal suspensions have been extensively studied experimentally and numerically in the past, as discussed below.

1.1.3.1 Experimental Studies

While early experimental studies focused primarily on spherical particles, recent investigations have highlighted the significant influence of particle shape on suspension dynamics. For example, studies of a swarm of settling particles have shown that the particle shape strongly influences the average bulk settling velocity [20–23]. In addition, shape characteristics, particularly angularity and aspect ratio, play a key role in influencing shear thickening and shear jamming behavior in dense suspensions of non-Brownian particles, i.e., elliptical, paraboloidal, and boat-shaped particles exhibit significantly stronger shear thickening than spherical particles due to their robust frictional contact network [24]. Studies of filtration processes have also highlighted the influence of particle shape, as the removal efficiency of cubic magnesium oxide particles was found to be much lower than that of the aerodynamically similar spheres [25].

In the effort to quantify shape-dependent behavior, precision in the creation and measurement of particle shapes is paramount to obtaining reliable and quantitative results. However, experimental efforts face significant hurdles in achieving this precision due to the inherent complexities involved. Controlling variables, evaluating results, and accurately determining parameters such as particle shape are formidable challenges [26]. In contrast, numerical studies offer a cost-effective and efficient alternative that allows tracking individual particles and precise control of shape parameters. They are therefore essential for gaining a deeper understanding of the underlying dynamics.

1.1.3.2 Numerical Simulations

Two main approaches are used to simulate particle-fluid systems: the Euler–Euler approach and the Euler–Lagrange approach. Both approaches treat the fluid as a continuum and generally use computational fluid dynamics (CFD) to solve the equations governing fluid flow.

The Euler–Euler approach typically relies on the advection-diffusion equation as the primary governing equation for the particle component and is thus computationally efficient [27, 28]. However, this approach has a limited scope because it considers particles and fluid as interpenetrating continua, which ignores the discrete nature of the dispersed phase and makes it best suited for monodisperse systems where particles are uniform. The particle-particle and particle-fluid interactions are only modeled and not locally resolved, making the approach unable to accurately capture the particle shape to characterize its effect on the behavior of the system.

The Euler–Lagrange approach, on the other hand, treats the particles as discrete entities, with their motion and interactions being tracked individually using methods such as the discrete element method (DEM). As a result, it is better suited to capture the intricate details of suspension dynamics. However, this advantage comes with a significant disadvantage: as the number of particles considered increases, so does the computational effort [29].

To further evaluate existing DEM approaches, it is essential to categorize particle-fluid systems based on the level of coupling, by distinguishing between three types: one-way, two-way, and four-way coupling. One-way coupling refers to the situation where the particles are affected by the fluid, but the fluid is not affected by the particles. This approach is suitable for dilute particle suspensions with low particle volume fractions. Two-way coupling takes into account the mutual interaction between the particles and the fluid. This level of coupling is necessary when the volume fraction of particles is moderate and their presence significantly affects the fluid flow. Four-way coupling is the most comprehensive, taking into account the mutual interaction between the particles and the fluid, as well as the interaction between the particles themselves and solid obstacles. This level of coupling is essential for particle-laden flows with high particle volume fractions, i.e. dense suspensions, where particle-particle and particle-wall interactions play a significant role in the overall system behavior. In general, as the volume fraction increases, a higher level of coupling is required [30]. Since dense suspensions are common in many applications [31], four-way coupling is a requirement to accurately model them.

Numerical simulations of particulate flows involving a large number of particles (> 1000) often employ subgrid scale (SGS) particle models. Traditionally, this approach treats the unresolved particles as ideal spheres and models particles as mass points governed by applied forces, such as drag, according to Newton's second law [32, 33]. To accommodate more complex shapes, extensions using ellipsoidal [34–37] and polygonal [38–40] particles have been developed. However, as the complexity of the particle shapes increases, so does the computational complexity of the simulations. More sophisticated approaches approximate arbitrary shapes by combining convex elements, using

Table 1.1: *Overview of publications in which fully resolved particles are considered in 3D, with the number of particles considered in the simulations and their shape.*

Reference	Year	Particle number	Shape
Fu et al. [49]	2021	125	Ellipsoids
Rosemann et al. [50]	2020	200	Spherocylinders
Uhlmann and Doychev [51]	2014	15190	Spheres
Walayat et al. [52]	2019	500	Ellipsoids
Willen et al. [53]	2017	2000	Spheres
Willen and Prosperetti [54]	2019	2000	Spheres
Yao et al. [55]	2021	2000	Spheres
Zaidi et al. [56]	2015	764	Spheres

multiple spheres [41, 42] or other more complex convex shapes [43–45]. These methods compute and aggregate the forces acting on each element to characterize the behavior of the entire particle, but accurately representing complex shapes requires a significant number of elements, leading to substantial computational demands. SGS simulations also suffer from inaccuracies in fluid-particle interactions, especially for non-spherical particles. This is because most SGS approaches use simplified spherical geometries [46, 47] or omit back-coupling to the fluid phase and only consider shape-simplifying drag correlations [48], which leads to significant errors, especially for moderate and high particle volume fractions [30].

In comparison, direct numerical simulations (DNS) have the advantage of resolving the flow around the particles, taking the actual geometry of the surface into account. However, only simple particle shapes are usually considered, i.e. without edges and corners, as shown in Table 1.1, but there are some models applicable to arbitrarily shaped particles, as introduced below.

The immersed boundary method (IBM) represents particle surfaces using a discrete set of Lagrangian points within an Eulerian mesh [57, 58], making it a viable choice for simulating particles of any shape. A key strength of the IBM lies in its exceptional flexibility. It can be easily paired with a variety of fluid solvers, such as finite element methods and the lattice Boltzmann method (LBM). It is also highly accurate because the interaction between the Lagrangian points and the fluid mesh is independent. Furthermore, IBM offers the possibility to consider four-way coupling with appropriate contact models for arbitrary shapes [59]. However, a notable limitation of IBM is that it requires a sufficiently fine mesh to accurately capture interactions between the Eulerian mesh and

Lagrangian points through interpolation. This requirement results in high computational costs, especially when combined with advanced collision detection algorithms [37, 60].

The partially saturated method (PSM) [61], an alternative LBM-based approach, and its subsequent derivatives have been widely used in the investigation of particulate flows [62, 63]. For example, to study the filtration process of irregular airborne particles [64]. The increased preference for PSM is consistent with the growing interest in LBM, which is known for its computational efficiency and ease of parallelization. This is primarily due to the fact that LBM's computationally intensive operations are inherently local [65].

The homogenized lattice Boltzmann method (HLBM) [66, 67] can be seen as another derivative of PSM [62]. This monolithic approach, capable of simulating fluid, porous media, and particle-laden flows, has been chosen for the comprehensive numerical investigation of shape-dependent single particle settling [68]. However, the application of this method and other PSM derivatives to dense suspensions requires the introduction of a compatible explicit contact model [69]. In addition, although the semi-local algorithm is highly compatible with massive parallelization [70] and methods exist for distributing pure fluid and fluid-particle interaction computations among processing units (PUs), no such methods currently exist for pure particle computations. This leads to an immense communication and computational overhead, since the latest particle data must be available to all PUs in order to perform identical computations. The overhead increases with the number of particles considered, further limiting its current applicability to dense suspensions.

In summary, existing methods are unable to effectively simulate hundreds or even thousands of complexly shaped particles in four-way coupled systems. The task is particularly challenging because of the need to preserve the details of the shapes while keeping the resulting computational effort within reasonable limits.

1.2 Challenges and Aims

Current research focuses mainly on analyzing simple shapes. When complex shapes are considered, studies tend to be limited in scope, examining only a small number of particles, or the findings often lean toward qualitative descriptions rather than quantitative assessments. This trend underscores a significant gap in the literature: the need for comprehensive quantitative analysis of complex shapes in particulate flows. Addressing this gap is crucial for advancing the understanding and modeling of these systems, which are prevalent in various real-world applications. Hence, the development and implementation of numerical models tailored to study complex real-world problems is imperative.

Based on the identified gap in the current literature, the primary objective of this thesis formulated as follows:

> *The main goal of this work is to develop and validate, using analytical solutions and benchmarks from literature, a novel model capable of capturing the dynamics of suspensions with high particle volume fractions (up to 30%), while allowing arbitrary convex particle shapes.*

To this end, HLBM is adopted due to its potential for handling particulate flows involving arbitrarily-shaped particles. However, current limitations such as the communication and computational overhead and the lack of a compatible explicit contact model must be addressed. The following subgoals are therefore essential to reach the main objective, demonstrate its feasibility, and ensure the sustainability of the resulting software:

Subgoal 1: Improvement in parallel performance by reducing the communication and computational overhead of particle-related operations in fully resolved simulations of particle-laden flows.

Subgoal 2: Introduction of a versatile, nonlinear contact model for complex convex shapes, seamlessly integrated with HLBM to enable four-way coupled particulate flow simulations.

Subgoal 3: The resulting software implementation is reusable, freely available, and easily extensible.

Subgoal 4: Detailed numerical studies of hindered settling using spherical and non-spherical particle shapes to demonstrate applicability to capture suspension dynamics with complex particle shapes.

By addressing these subgoals, this research aims to provide a robust and versatile tool for the quantitative analysis of complex particulate flows, thereby filling a critical gap and advancing the field.

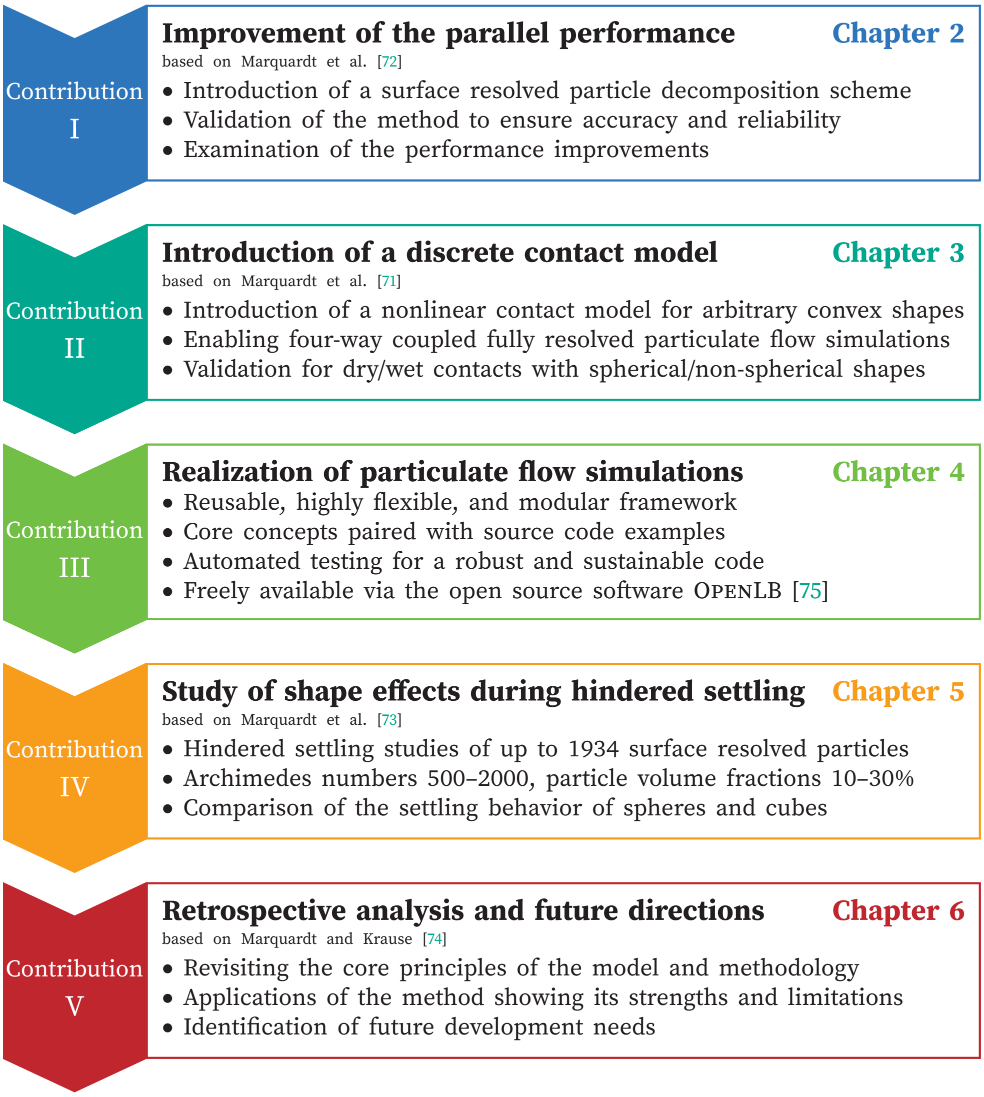
1.3 Thesis Structure

To achieve the above goals, this thesis presents five key contributions derived from four dedicated works [71–74]. An overview of these contributions is provided in Figure 1.1, with detailed explanations following below. Section B of the appendix lists all simulation setups used in this work, ensuring full reproducibility.

Chapter 2, which is based on Marquardt et al. [72], covers **Contribution I — Improvement of the parallel performance**. It addresses the performance challenge, i.e. the communication and computational overhead when simulating (large numbers of) fully resolved particles (Subgoal 1). To address this issue, it introduces the fundamentals of HLBM and proposes a compatible surface resolved particle decomposition scheme. This scheme aims to improve scalability by optimizing inter-process communication, data management, and workload distribution across the PUs, ultimately improving overall simulation performance. The performance improvement and scalability with respect to the number of particles and the resolution of the computational domain are evaluated and the correctness of the proposed method is validated.

In Chapter 3, which is based on Marquardt et al. [71], a novel discrete contact model for complex convex particle shapes is introduced (Subgoal 2), marking **Contribution II — Introduction of a discrete contact model**. This nonlinear contact model is compatible with HLBM and PSM, thus enabling four-way coupled fully resolved particulate flow simulations. Essential contact properties, such as indentation depth, are obtained by discretizing the contact region. To assess its validity, extensive tests are conducted, including scenarios with and without fluid, as well as spherical and non-spherical particle shapes. The results of these tests are then compared with analytical solutions and results from literature.

Core concepts and implementation details of a reusable, highly flexible, and modular software framework (**Contribution III — Realization of particulate flow simulations**) designed to perform efficient and accurate particulate flow simulations (Subgoal 3) are given in Chapter 4. The framework is designed to separate data and operations, allowing for easy customization and extension, making it highly flexible. The versatile particle data structure allows for effortless modification of particle properties. This modularity simplifies the process of adding or removing particle properties as needed, making the software a powerful and adaptable tool, for example, it can be utilized for both fully resolved and subgrid scale particles. However, the primary focus of this work is



Contribution I

Improvement of the parallel performance **Chapter 2**

based on Marquardt et al. [72]

- Introduction of a surface resolved particle decomposition scheme
- Validation of the method to ensure accuracy and reliability
- Examination of the performance improvements

Contribution II

Introduction of a discrete contact model **Chapter 3**

based on Marquardt et al. [71]

- Introduction of a nonlinear contact model for arbitrary convex shapes
- Enabling four-way coupled fully resolved particulate flow simulations
- Validation for dry/wet contacts with spherical/non-spherical shapes

Contribution III

Realization of particulate flow simulations **Chapter 4**

- Reusable, highly flexible, and modular framework
- Core concepts paired with source code examples
- Automated testing for a robust and sustainable code
- Freely available via the open source software OpenLB [75]

Contribution IV

Study of shape effects during hindered settling **Chapter 5**

based on Marquardt et al. [73]

- Hindered settling studies of up to 1934 surface resolved particles
- Archimedes numbers 500–2000, particle volume fractions 10–30%
- Comparison of the settling behavior of spheres and cubes

Contribution V

Retrospective analysis and future directions **Chapter 6**

based on Marquardt and Krause [74]

- Revisiting the core principles of the model and methodology
- Applications of the method showing its strengths and limitations
- Identification of future development needs

Figure 1.1: *Overview of the thesis structure, illustrating the division into different phases and their corresponding chapters.*

on fully resolved particles. In this context, the framework integrates the HLBM with the novel particle decomposition scheme and the discrete contact model, as outlined in the previous steps. By combining these components, the software is able to study suspension dynamics. The framework is subjected to automated testing, which helps ensure its robustness and sustainability. Furthermore, this framework is freely available via the open source software OpenLB [75].

Contribution IV — Study of shape effects during hindered settling, presented in Chapter 5, is the application of the DNS approach to the hindered settling of volume-equivalent spheres and cubes with up to 1934 surface resolved particles (Subgoal 4). The chapter is based on Marquardt et al. [73] and evaluates the grid independence, the convergence of the method, and validates the model by comparing the obtained results for spherical particles with correlations from literature, to ensure the reliability and accuracy of the results. The effect of particle shape on the settling behavior of the spheres and cubes is also investigated over a range of Archimedes numbers (500–2000) and particle volume fractions (10–30%). The results of this study aim to provide insights into the role of particle shape in non-colloidal suspension behavior and to demonstrate the effectiveness of the novel HLBM-based DNS approach for studying suspension dynamics.

The final contribution (**Contribution V — Retrospective analysis and future directions**) in Chapter 6, is a comprehensive concluding review. This chapter, based on the work of Marquardt and Krause [74], revisits the underlying models and methods. Through an additional exploration of various reported applications, the chapter demonstrates the versatility and robustness in addressing complex particulate flow problems, including but not limited to suspension dynamics. It highlights the strengths and capabilities and identifies potential areas for future development to improve the applicability of the method to an even broader range of scientific and engineering disciplines.

Finally, Chapter 7 provides a comprehensive conclusion by summarizing the key takeaways presented in the previous chapters and an outlook on potential future research.

2
Discrete Particulate Flow Model

This chapter has been published as

Marquardt, J. E. et al. "A novel particle decomposition scheme to improve parallel performance of fully resolved particulate flow simulations." In: Journal of Computational Science *78 (2024), p. 102263.* ISSN*: 1877-7503.* DOI*: 10.1016/j.jocs.2024.102263*

2.1 Introduction

In numerous industrial and scientific contexts, the study of particulate systems has emerged as a critical area of research. Particles have an astonishing variety of complex shapes, making their behavior highly complex and difficult to understand. Understanding and manipulating the behavior of these complexly shaped particles is vital to improving existing industrial processes and developing new technological advances.

The extensive experimental and numerical research devoted to the study of hindered settling highlights the importance of studying particle collectives. Research initially focused on obtaining correlations for the average settling velocity in order to establish relationships between suspension characteristics and settling behavior [76–80]. These early studies provided valuable insights into the collective behavior of particles and laid the foundation for further investigations of dilute [81] and concentrated suspensions at moderate Reynolds numbers [56, 82]. Subsequently, the research expanded to include the formation and dynamics of particle clusters during hindered settling [54, 55]. Understanding the clustering is crucial as it significantly influences the overall settling behavior and the efficiency of separation processes.

Despite these extensive research efforts, it is important to note that the above studies have focused solely on simple spherical geometries. However, in many industrial and scientific applications, particles have non-spherical and usually irregular shapes. Therefore, several experimental studies consider more complex shapes, such as cubic and brick-like shapes [20], fibers [23], and irregular sand grains [22]. Current knowledge, however, is limited to explicit shapes, and a deeper understanding, including correlations of the average settling velocity with shape factors, remains a challenge. Experiments, though valuable, are time-consuming, costly, and inherently difficult to control and adjust specific parameters, such as particle shapes. In addition, it is difficult to resolve a large number of particles and high particle volume fractions.

Numerical investigations therefore play an essential role in gaining a thorough understanding of such systems. However, comprehensive models are complex and computationally expensive due to the need to incorporate four-way coupling. This includes coupling between fluid and particles, between particles themselves, and between particles and walls. It is essential to accurately capture the complex interactions and dynamics within the system. In addition, the presence of complex shapes introduces shape factors that affect collective behavior. Hence, to obtain statistically meaningful results, a larger number of particles is required, which further increases computational demands.

The discrete element method (DEM) is one approach to consider particulate flows. It is widely used for simulations in many fields [33] and can represent arbitrary geometries using the glued-sphere technique [41] or similarly by combining other convex shapes [44]. However, approximating arbitrary shapes from combinations results in limited accuracy. To improve the accuracy, the number of segments must be increased significantly, resulting in expensive computations. In addition, modeling the influence of the surrounding fluid is challenging. While there have been studies on coupling with a surrounding fluid, these mainly use simple spherical geometries [46, 47] or model only the drag coefficient without back-coupling from the particles to the fluid [48]. However, the back-coupling is of enormous importance, especially at high particle volume fractions [30].

The immersed boundary method (IBM) is another well-known option to consider non-spherical particles, since it resolves surfaces as Lagrange points [58]. The interaction of these points with the fluid is independent of the fluid grid, resulting in a high level of accuracy. In addition, a significant advantage of IBM is its ability to be coupled with various fluid solvers, including the finite element method and the lattice Boltzmann method (LBM). For example, it has been used to study the settling behavior of elliptical particles [83, 84], which has also been addressed by other LBM-based methods [85]. However, the frequent interpolations between particle and fluid points are computationally expensive.

The partially saturated method (PSM), originally introduced by Noble and Torczynski [61] is the most widely used alternative LBM-based method for simulating particles of arbitrary shape. This preference for PSM is consistent with the growing interest in LBM due to its efficiency and ease of parallelization, driven by the fact that its computationally intensive operations are inherently localized [65]. Subsequently, various new derivatives have been introduced [62, 63], including the homogenized lattice Boltzmann method (HLBM) [66, 67, 69]. The ability to accurately represent a wide range of shapes has been demonstrated in several cases. For example, this capability has been used to derive shape-dependent drag coefficient correlations [68]. In addition, HLBM has been applied to the simulation of cubic disk-shaped particles in wall flow filters [86–88] and another PSM derivative to the filtration of irregular airborne particles [64]. The potential applications are further extended by a compatible contact model that is suitable for arbitrary convex shapes [71]. Since this feature enables four-way coupled simulations using PSMs that already take advantage of the LBM's parallelization capabilities to improve the particle-fluid coupling performance. However, the common implementation of PSMs involves communicating all data to each process involved in the computation, which can lead to significant overhead and limit the scalability of simulations, especially at higher particle counts and domain sizes.

In summary, direct numerical simulations are essential for the numerical study of suspensions consisting of arbitrarily shaped particles, and several approaches, such as PSM and HLBM, are available for such simulations. However, these approaches share a significant computational cost, which poses a challenge in their application to more realistic scenarios involving hundreds or even thousands of particles. Consequently, there is an urgent need to increase their efficiency.

The aim of the present work is therefore to improve the efficiency of PSMs through the development of a novel particle decomposition scheme that allows for more efficient simulations. By achieving this objective, future PSM simulations are conductable with larger particle populations, allowing for more accurate and comprehensive investigations of complex-shaped particle systems.

This eventually increases its applicability to real-world problems.

To this end, the remainder of this paper is organized as follows. Section 2.2 introduces the models used to consider the fluid and the particles, while Section 2.3 discusses the numerical methods used to solve the model system, followed by the proposed domain decomposition scheme in Section 2.4 and its application to hindered settling in Section 2.5. Finally, Section 2.6 provides a summary and conclusion.

2.2 Modeling

2.2.1 Fluid

Fluids are commonly considered to be incompressible. In this case, the Navier–Stokes equations become

$$\begin{aligned} \frac{\partial \boldsymbol{u}_\mathrm{f}}{\partial t} + (\boldsymbol{u}_\mathrm{f} \cdot \nabla)\, \boldsymbol{u}_\mathrm{f} - \nu \Delta \boldsymbol{u}_\mathrm{f} + \frac{1}{\rho_\mathrm{f}} \nabla p = \boldsymbol{F}_\mathrm{f}, \\ \nabla \cdot \boldsymbol{u}_\mathrm{f} = 0, \end{aligned} \tag{2.1}$$

where p is the pressure, t is the time, $\boldsymbol{F}_\mathrm{f}$ is the total of all forces acting on the fluid, and $\boldsymbol{u}_\mathrm{f}$, ρ_f, ν are the velocity, density, and kinematic viscosity of the fluid.

2.2.2 Particle

For the particle component, we use Newton's second law of motion. Therefore, translation is described by

$$m_\mathrm{p} \frac{\partial \boldsymbol{u}_\mathrm{p}}{\partial t} = \boldsymbol{F}_\mathrm{p} \tag{2.2}$$

and rotation by

$$\boldsymbol{I}_\mathrm{p} \frac{\partial \boldsymbol{\omega}_\mathrm{p}}{\partial t} + \boldsymbol{\omega}_\mathrm{p} \times (\boldsymbol{I}_\mathrm{p} \cdot \boldsymbol{\omega}_\mathrm{p}) = \boldsymbol{T}_\mathrm{p}. \tag{2.3}$$

Here, m_p, $\boldsymbol{I}_\mathrm{p}$, $\boldsymbol{u}_\mathrm{p}$, $\boldsymbol{\omega}_\mathrm{p}$ are the mass, moment of inertia, velocity, and angular velocity of the particle. $\boldsymbol{F}_\mathrm{p}$ and $\boldsymbol{T}_\mathrm{p}$ are the sum of the acting forces and torques affecting the particle motion, which may include the hydrodynamic forces and torques mentioned in Section 2.3.2 or contact treatment results [71, 89]. Above, the subscript p indicates that the quantities refer to the particle's center of mass.

2.3 Numerical Methods

2.3.1 Lattice Boltzmann Method

In this work, the Navier–Stokes equations for incompressible flows are solved using the lattice Boltzmann method (LBM) [65, 90, 91]. Note that all values in Section 2.3 are given in lattice units, unless explicitly stated otherwise.

LBM has its roots in gas kinetics, which explains why it operates at the mesoscopic level and considers the behavior of particle populations. Accordingly, particles in this section refer to the fluid particles. The discrete velocity distribution function $f_i(\boldsymbol{x}, t)$ is used to characterize the aforementioned populations at a position $\boldsymbol{x}$ and time t. The index i refers to the corresponding discrete velocity $\boldsymbol{c}_i$, which is given by the selected velocity set. There are several velocity sets available in the literature [65, 90]. For the studies in this paper, we choose the D3Q19, which discretizes the three-dimensional space and contains 19 discrete velocities

$$\boldsymbol{c}_i = \begin{cases} (0,0,0), & \text{if } i = 0 \\ (\pm 1,0,0),(0,\pm 1,0),(0,0,\pm 1), & \text{if } i = 1,\dots,6 \\ (\pm 1,\pm 1,0),(\pm 1,0,\pm 1),(0,\pm 1,\pm 1), & \text{if } i = 7,\dots,18 \end{cases} . \tag{2.4}$$

The populations are also used to derive macroscopic quantities such as the fluid density $\rho_\mathrm{f}(\boldsymbol{x}, t) = \sum_i f_i(\boldsymbol{x}, t)$ and velocity $\rho_\mathrm{f}\boldsymbol{u}_\mathrm{f}(\boldsymbol{x}, t) = \sum_i \boldsymbol{c}_i f_i(\boldsymbol{x}, t)$.

The particle populations' evolution over time is expressed by the lattice Boltzmann equation that is usually divided into a collision and streaming step. The former reads

$$f_i^*(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) + \Omega_i(\boldsymbol{x}, t) + S_i(\boldsymbol{x}, t). \tag{2.5}$$

Here, the post-collision distribution f_i^* is obtained using a collision operator Ω_i and an optional source term S_i. Furthermore, the propagation step with $\Delta t = \Delta x = 1$ is given by

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i \Delta t, t + \Delta t) = f_i^*(\boldsymbol{x}, t), \tag{2.6}$$

which streams the particle populations to their corresponding neighboring lattice nodes.

The simplest way to account for collisions is to relax the distributions toward their equilibrium f_i^eq, as is done by the Bhatnagar–Gross–Krook (BGK) collision operator [92]

$$\Omega_i(\boldsymbol{x}, t) = -\frac{1}{\tau}(f_i(\boldsymbol{x}, t) - f_i^\mathrm{eq}(\rho_\mathrm{f}, \boldsymbol{u}_\mathrm{f})), \tag{2.7}$$

with the relaxation time τ that determines the speed of the relaxation and depends on the kinematic viscosity ν as follows

$$\tau = \left(3\nu + \frac{1}{2}\right). \tag{2.8}$$

The Maxwell–Boltzmann distribution, quantifying the equilibrium state is given by [93]

$$f_i^{\mathrm{eq}}(\rho_{\mathrm{f}}, \boldsymbol{u}_{\mathrm{f}}) = w_i \rho_{\mathrm{f}} \left(1 + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}_{\mathrm{f}}}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}_{\mathrm{f}})^2}{2c_s^4} - \frac{\boldsymbol{u}_{\mathrm{f}} \cdot \boldsymbol{u}_{\mathrm{f}}}{2c_s^2}\right). \tag{2.9}$$

The required weights w_i originate from a Gauss-Hermite quadrature rule and are fixed for the chosen velocity set, as is the constant lattice speed of sound c_s. For D3Q19, the weights read

$$w_i = \begin{cases} 1/3, & \text{if } i = 0 \\ 1/18, & \text{if } i = 1, \ldots, 6 \\ 1/36, & \text{if } i = 7, \ldots, 18 \end{cases}, \tag{2.10}$$

and the lattice speed of sound is given by $c_s = 1/\sqrt{3}$.

The lattice Boltzmann method (LBM) implemented in the open source software OpenLB [75, 94] is used exclusively in this work.

2.3.2 Homogenized Lattice Boltzmann Method

Although the proposed scheme is potentially applicable to a variety of methods such as PSM, in the context of this work, we consider the application to HLBM [66, 67, 69]. Using the trigonometric level set function [71]

$$B(\boldsymbol{x}, t) = \begin{cases} 1, & \text{if } d_s \leq -\varepsilon/2 \\ \cos^2\left(\frac{\pi}{2}\left(\frac{d_s}{\varepsilon} + \frac{1}{2}\right)\right), & \text{if } \varepsilon/2 > d_s > -\varepsilon/2 \\ 0, & \text{if } d_s \geq \varepsilon/2 \end{cases}, \tag{2.11}$$

particles are mapped onto the entire computational domain for later coupling between the components, with the size of the smooth boundary ε and the signed distance to the boundary d_s. Inside the object, the distance is negative, outside it is positive. In the remainder of the paper, we use $\varepsilon = 1/2$, in accordance with Krause et al. [66]. The coupling from the particle to the fluid is based on a velocity difference obtained by a convex combination of the fluid and particle velocities

$$\Delta \boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x}, t) = B(\boldsymbol{x}, t)\big(\boldsymbol{u}_{\mathrm{p}}(\boldsymbol{x}, t) - \boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x}, t)\big), \tag{2.12}$$

where $\boldsymbol{u}_\mathrm{p}(\boldsymbol{x}) = \boldsymbol{u}_\mathrm{p}(\boldsymbol{X}_\mathrm{p}) + \boldsymbol{\omega}_\mathrm{p} \times (\boldsymbol{x} - \boldsymbol{X}_\mathrm{p})$ is the particle's velocity at position $\boldsymbol{x}$, and $\boldsymbol{X}_\mathrm{p}$ is the particle's center of mass. According to previous studies [69], the best results are obtained by using an adapted exact difference method (EDM) [95]. This method introduces the following source term in Eq. (2.5)

$$S_i(\boldsymbol{x}, t) = f_i^{\mathrm{eq}}(\rho_\mathrm{f}, \boldsymbol{u}_\mathrm{f} + \Delta \boldsymbol{u}_\mathrm{f}) - f_i^{\mathrm{eq}}(\rho_\mathrm{f}, \boldsymbol{u}_\mathrm{f}). \tag{2.13}$$

To couple from the fluid to the particle, we use the momentum exchange algorithm (MEA) by Wen et al. [96] to calculate the hydrodynamic force

$$\boldsymbol{F}_\mathrm{h}(\boldsymbol{x}, t) = \sum_i (\boldsymbol{c}_i - \boldsymbol{u}_\mathrm{p}(\boldsymbol{x}, t)) f_i(\boldsymbol{x} + \boldsymbol{c}_i \Delta t, t) + (\boldsymbol{c}_i + \boldsymbol{u}_\mathrm{p}(\boldsymbol{x}, t)) f_{\bar{i}}(\boldsymbol{x}, t). \tag{2.14}$$

Above, the index $\bar{i}$ refers to particle populations with the corresponding velocity $\boldsymbol{c}_{\bar{i}} = -\boldsymbol{c}_i$, i.e. the population $f_{\bar{i}}$ points in the opposite direction of f_i.

The sum of all hydrodynamic forces of cells whose center is inside the particle is now the total hydrodynamic force acting on the particle. These cells are denoted by $\boldsymbol{x}_\mathrm{b}$. Thus the total force is [69]

$$\boldsymbol{F}_\mathrm{p}(t) = \sum_{\boldsymbol{x}_\mathrm{b}} \boldsymbol{F}_\mathrm{h}(\boldsymbol{x}_\mathrm{b}, t), \tag{2.15}$$

and the torque is given by

$$\boldsymbol{T}_\mathrm{p}(t) = \sum_{\boldsymbol{x}_\mathrm{b}} (\boldsymbol{x}_\mathrm{b} - \boldsymbol{X}_\mathrm{p}) \times \boldsymbol{F}_\mathrm{h}(\boldsymbol{x}_\mathrm{b}, t). \tag{2.16}$$

Note that while an applicable contact model exists [71], we deliberately refrain from using it in the context of this work in order to focus exclusively on studying the performance improvement of the decomposition scheme.

2.4 Particle Decomposition Scheme

2.4.1 Background

In the context of the LBM, parallelization is typically implemented using a block-based approach [90]. This involves dividing the computational domain into multiple blocks, with each block being assigned to a specific process unit. In this parallel scheme, a process unit can handle multiple blocks or a single block, depending on the workload distribution.

In the LBM framework, data is stored locally for each block, allowing for efficient computations within each process. However, as the boundaries of the blocks interact, it becomes necessary to communicate data across these boundaries with directly adjacent blocks. Implementations of PSMs follow this approach. The calculation of the surface forces is done locally using the fluid information available within each block. However, once calculated, these surface forces are communicated to all other processes involved in the simulation for summation. This allows synchronized solving of the equations of motion on each processor.

2.4.2 Improvements

This section introduces three new steps necessary for particle decomposition, which allow for more efficient communication than the communication between all processes involved as described above. Importantly, these steps do not change the underlying methodology. Rather, they balance the workload over multiple processes and ensure data consistency without affecting the accuracy or stability of the method. They are also applicable to any shape, since the only geometric parameter they depend on is the circumferential radius. The steps include the communication of surface forces and torques, the assignment of particles, and the communication of particle data. The following sections provide basic definitions and a detailed explanation of each step and its relevance.

2.4.2.1 Definitions

To enhance the comprehensibility, we present several definitions that we use in the following sections:

Responsibility When a block or its corresponding process unit assumes responsibility for a particle, it means that both the solving of the equations of motion and the subsequent reassignment of the particle are performed within that specific process.

Neighborhood The neighborhood refers to the collection of blocks located within the maximum circumferential radius, which is the largest radius of all particles in the simulation, from the block of interest. Neighboring processes correspond to the processes responsible for handling the blocks within this defined block neighborhood.

Extended neighborhood The extended neighborhood in this context includes neighboring blocks and their neighboring blocks, creating an extended spatial region that includes not only immediate neighbors but also the secondary level of neighboring blocks.

2.4.2.2 Communication of Local Fluid to Particle Coupling Results

We use a communication-ideal strategy, i.e., particle data is stored and used for coupling on the process units that also know the associated fluid data, due to the advantages outlined by Henn et al. [97]. Therefore, effective communication of surface forces and torques between process units is required to ensure accurate simulations. Due to the decomposition of the fluid domains across processors, the responsible processor may lack essential fluid data for the surface parts of a particular particle, as illustrated in Figure 2.1. In such scenarios, portions of the force and torque are missing and are provided by other processes that have the relevant data, analogously to the conventional implementation.

The new and improved data communication process involves two types of processors: senders and receivers. The sender processor is the one that holds part of the surface of a particle, but is not responsible for it. On the other hand, the receiver processor is responsible for the particle in question. In Figure 2.1, the former is highlighted with a blue cross-hatch and the latter with a brown fill.

The communicated data includes the particle ID, which uniquely identifies the particle within the simulation. In addition, the sender transmits the partial surface force, which represents the force acting on the surface part of the particle held by the sender. The torque resulting from the partial surface force is also included.

2.4.2.3 Particle Assignment

The assignment of particles to processors is done at the block level, similar to the fluid domain decomposition. This means that a single process may handle multiple particle blocks, depending on the computational domain and decomposition scheme. After solving the equations of motion, the positions of the particles may have changed. Consequently, the assignment of particles to processors must be reevaluated.

The responsibility for a particle goes to the block on which the center of mass of the particle is located. Additionally, all neighboring blocks are assumed

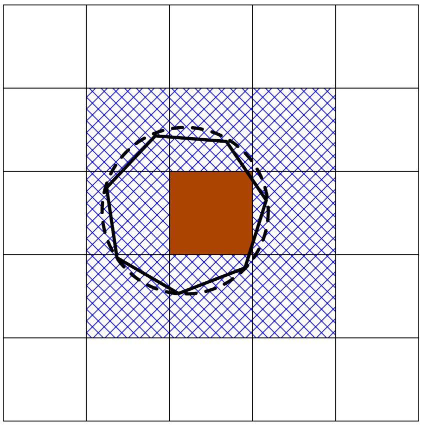


Figure 2.1: *Representation of a heptagonal particle (solid black) and its circumferential radius (dashed black) in a multi-block domain consisting of the responsible center block (brown) and neighboring blocks (blue cross-hatching).*

to touch the surface of the particle. These neighboring blocks calculate the partial surface forces and torques, but are not individually responsible for the particle itself. On the other hand, blocks that are not in the neighborhood are not assigned to the particle of interest. The responsible block described above is shown in Figure 2.1 with a brown fill, the neighboring blocks with a blue crosshatch, and all other blocks without fill. This selectivity ensures an efficient use of computational resources by involving only the necessary blocks in particle interaction calculations and minimizing the following communication overhead by excluding as many process units as possible.

Note that using the maximum circumferential radius to determine the neighborhood for particle assignment avoids missing overlaps and thus ensures data consistency in the communication steps, no matter how complex and different the shapes in the particle system are.

2.4.2.4 Communication of Particle Data

This communication step, which occurs after the particle reassignment, is essential to maintain data consistency and involves more blocks, as shown in Figure 2.2. The figure contains a heptagonal particle at the current position and at the next time step, indicated by the dashed and solid lines, respectively. There are also several types of blocks: the responsible block at the current time step (brown), the responsible block at the next time step (brown with blue cross-hatch), the current neighborhood (blue cross-hatch), and the additional

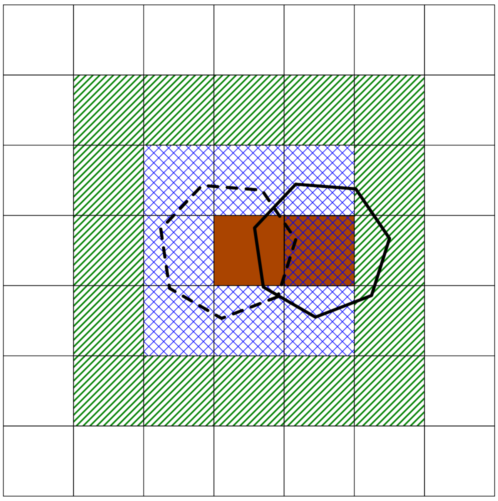


Figure 2.2: *Illustration of the position of a heptagonal structure at the current (dashed black) and next (solid black) time steps within a multi-block domain. The brown middle block represents the currently responsible block, while the adjacent blocks with blue cross-hatching indicate the first-level neighborhood. The highlighted block with a combination of brown fill and blue cross-hatching shows the responsible block at the next time step. The green hatching represents the secondary neighborhood.*

blocks in the current extended neighborhood (green hatching).

The responsible block initiates a request to its entire extended neighborhood to ensure that no surface forces are overlooked due to potential changes in particle responsibility, because, as shown in Figure 2.2, parts of the particle surface may intersect with the extended neighborhood at the next time step. While the neighborhood of the block where the particle's center of mass is located after reassignment receives the actual particle data, all other blocks receive an empty request, streamlining the communication process.

The data sent during the communication step includes all relevant particle information. This typically consists of the particle's unique ID, position, angle, velocity, angular velocity, an ID of the responsible block or process unit, a surface ID, and any other particle properties needed for the specific simulation.

If a block previously held a valid particle but receives no updates in this step, the particle is invalidated and the corresponding memory is released at a predefined interval.

Algorithm 2.1: Basic LBM time step algorithm using PSM with the particle decomposition scheme

```
for all time steps do
    Couple fluid to particles;                       ▷ Using the MEA
    Communicate surface forces and torques;    ▷ See Section 2.4.2.2
    Apply external forces;                        ▷ Such as gravity
    Solve equations of motion;
    Evaluate particle assignment;              ▷ See Section 2.4.2.3
    Communicate data and assignment;           ▷ See Section 2.4.2.4
    Couple particles to fluid;                       ▷ Using the EDM
    Perform collision and streaming;
    Increase time step;
end
```

2.4.2.5 Time Step Algorithm

To provide a clear and organized presentation, Algorithm 2.1 presents the basic time step algorithm of the LBM with PSM and the proposed particle decomposition as it is implemented in the open source software OpenLB [94]. The algorithm follows a specific sequence that is consistent with the previously discussed methods and ensures the necessary order of execution.

2.5 Application to Hindered Settling

In the following, we evaluate the application of the proposed scheme to hindered settling [76]. To ensure the correctness of the results, we validate the average settling velocity, for which many correlations are known. Most correlations, like the one proposed by Richardson and Zaki [77], use a power-law model such as

$$\bar{u}_\mathrm{p} = u^*(1-\phi_\mathrm{p})^n, \tag{2.17}$$

with the predicted average settling velocity $\bar{u}_\mathrm{p}$, a reference terminal velocity of a single particle in the considered domain u^*, the particle volume fraction ϕ_p

and an expansion index

$$n = \begin{cases} 4.65, & \text{if } Re < 0.2 \\ 4.35Re^{-0.03}, & \text{if } 0.2 \leq Re < 1 \\ 4.45Re^{-0.1}, & \text{if } 1 \leq Re < 500 \\ 2.39, & \text{if } 500 \leq Re \end{cases}. \tag{2.18}$$

Here, n depends on the Reynolds number $Re = u^* D_\mathrm{s}/\nu$. Garside and Al-Dibouni [79] suggest

$$n = \frac{5.1 + 0.27Re^{0.9}}{1 + 0.1Re^{0.9}}, \tag{2.19}$$

which has shown a superior accuracy [98]. Barnea and Mizrahi [78] propose a different model, in which the average settling velocity is independent of Re and is given by

$$\bar{u}_\mathrm{p} = u^* \frac{1 - \phi_\mathrm{p}}{(1 + \phi_\mathrm{p}^{1/3}) \exp \frac{5\phi_\mathrm{p}}{3(1-\phi_\mathrm{p})}}. \tag{2.20}$$

For validation purposes, all three correlations are used below.

In the creeping flow regime, the reference velocity reads

$$u^*_\mathrm{St} = \frac{g D_\mathrm{s}^2}{18\nu} \left(\frac{\rho_\mathrm{p} - \rho_\mathrm{f}}{\rho_\mathrm{f}} \right), \tag{2.21}$$

where $g = 9.80665\ \mathrm{m\,s^{-2}}$ is the standard gravity and D_s is the diameter of the considered spherical particle. Furthermore, ρ_f and ρ_p are the fluid and particle densities. However, in the following studies we only consider higher Reynolds numbers, hence we calculate the reference velocity using

$$u^* = \sqrt{\frac{4 g D_\mathrm{s}}{3 C_d} \left(\frac{\rho_\mathrm{p} - \rho_\mathrm{f}}{\rho_\mathrm{f}} \right)}, \tag{2.22}$$

with the drag coefficient C_d that we compute using the approximation by Schiller and Neumann [99]

$$C_d = \frac{24}{Re} \left(1 + 0.15Re^{0.687} \right), \tag{2.23}$$

which is valid for $Re < 800$.

As is common in studies of hindered settling [55, 98], we use two dimensionless parameters to describe the setup below. The first one is the particle to fluid density ratio defined as $\rho_\mathrm{p}/\rho_\mathrm{f}$ and the second is the Archimedes number

$$Ar = \frac{g D_\mathrm{s}^3 \frac{\rho_\mathrm{p}-\rho_\mathrm{f}}{\rho_\mathrm{f}}}{\nu^2}. \tag{2.24}$$

2.5.1 Simulation Setup

To numerically study the above, we employ spherical particles with a diameter $D_\mathrm{s} = 2$ mm at random positions in a cubic domain with an edge length of $12D_\mathrm{s}$ and periodic boundaries on each side, which is filled with a fluid with a density $\rho_\mathrm{f} = 1000\ \mathrm{kg/m^3}$. The setup is exemplified in Figure 2.3 with a particle volume fraction of about 0.2. The size of the domain was chosen because previous studies have shown that sufficiently accurate results can be obtained from a size of $10D_\mathrm{s}$ [55, 98]. However, in the context of this work, we intend to generate larger particle numbers for testing purposes, and therefore expand the domain, but only slightly, so that the simulation remains feasible for high resolutions.

The initially resting particles are accelerated in the z-direction by the force $F_g = (\rho_\mathrm{f} - \rho_\mathrm{p})V_\mathrm{s}g$. Here, $V_\mathrm{s} = \pi D_\mathrm{s}^3/6$ is the sphere's volume. As in previous numerical investigations of hindered settling [54, 55, 98], we apply a pressure gradient in the opposite direction of particle motion to obtain a net volume flow rate of zero. We calculate it using the Ergun equation [100] with a superficial velocity equal to the current average settling velocity $\bar{u}_\mathrm{p}$.

In our studies, the Archimedes number is fixed at $Ar = 1500$ and the density ratio at $\rho_\mathrm{p}/\rho_\mathrm{f} = 1.3$. From these two quantities, we derive the particle density as well as the fluid viscosity. To explore the effects of particle concentration, we vary the particle volume fraction in the range of 0.05 to 0.3.

Furthermore, we choose a constant lattice relaxation time of 0.55 for all simulations, while resolving the sphere's diameter with N cells, which differs in the following simulations. We simulate a total time of $300t^*$, with $t^* = D_\mathrm{s}/u^*_\mathrm{St}$. Due to the initial random packing, the startup time is significantly reduced, nonetheless, we start averaging the velocities to obtain $\bar{u}_\mathrm{p}$ after a time of $30t^*$, leaving a period of $270t^*$ that ensures statistically averaged results.

2.5.2 Grid Independence Study

In this section, we delve into a grid independence study using a particle volume fraction of $\phi_\mathrm{p} = 0.1$ by systematically exploring various grid resolutions

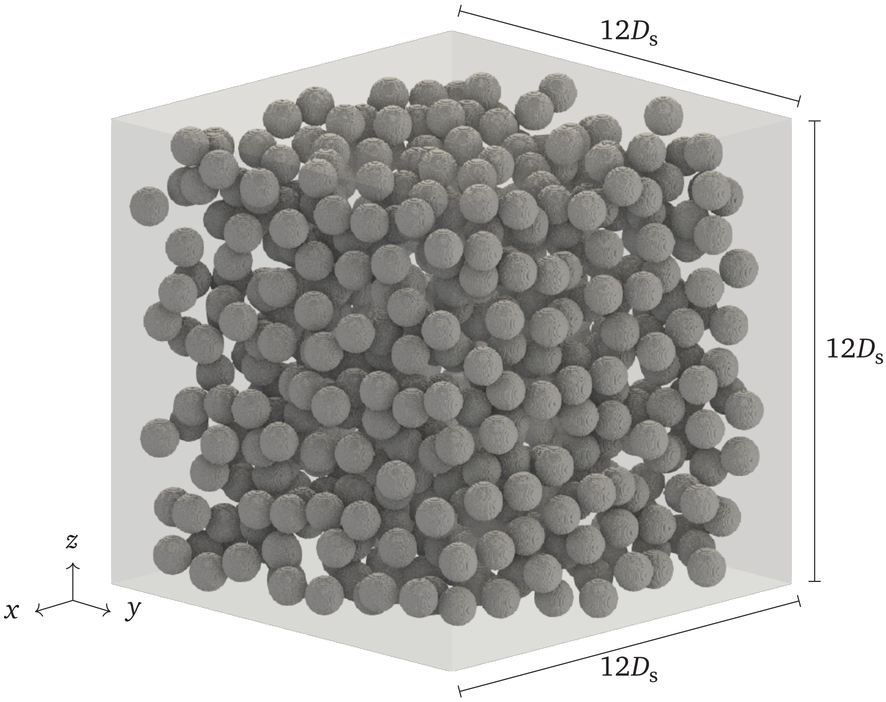


Figure 2.3: *Simulation domain with the spherical particles on their initial position using a resolution of* $N = 27$ *cells per sphere diameter and particle volume fraction of about* 0.2.

$N \in \{5, 7, 12, 18\}$. We compare the outcomes obtained from these resolutions to evaluate their impact on the simulation results, with respect to a baseline resolution of $N = 27$. Previous studies report sufficiently accurate results when resolving a sphere diameter with $N \approx 8$ cells [69]. In order to have a very accurate baseline, we therefore use a much finer resolution while still ensuring feasible simulations. The results are presented in Figure 2.4, which illustrates the relative error using the L^2 norm [90] plotted against the used grid resolutions. Additionally, the figure includes lines representing the experimental orders of convergence (EOC) with values of 1 and 2.

The plot reveals that as the grid resolution increases, the error of the simulation approximately follows the line of the EOC of 1, indicating a linear decrease in error with respect to grid refinement. The error decreases with increasing resolution as the particle shape is represented more accurately. The observed linear convergence is in agreement with the results of previous studies on single settling spheres [69] and suggests that increasing the grid resolution leads to

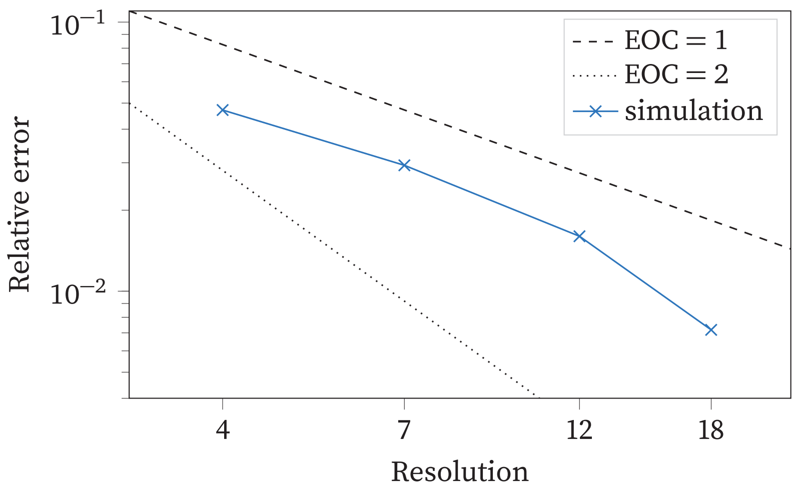


Figure 2.4: *Relative error in L^2 norm versus the resolution of the sphere's diameter N. The reference uses the resolution $N = 27$.*

a proportional decrease in error. It is likely due to the staircase approximation of the curved boundary [90, 101]. Moreover, it is noteworthy that the error becomes smaller than 2% starting from a resolution of $N = 12$, ensuring grid-independent results. The minimum resolution required is slightly higher than the resolutions reported in previous studies [69]. This adjustment is due to the presence of multiple particles and their interactions through the fluid.

2.5.3 Validation

The following validation uses the resolution $N = 18$, to assess the accuracy and reliability. To this end, we plot the average velocity obtained from the simulations over the particle volume fractions in Figure 2.5. For comparison, we additionally add the correlation by Richardson and Zaki [77] as a dotted purple line, Garside and Al-Dibouni [79] as a blue line, and the correlation by Barnea and Mizrahi [78] as a dashed green line. It is evident that as the particle volume fraction increases, the average settling velocity progressively decreases. In general, the results are well within the range of the correlations introduced above. In particular, they agree well with the first two correlations. However, it is noteworthy that at higher particle volume fractions, the disparity between the simulation results and correlations becomes more pronounced.

HLBM is therefore able to capture the settling behavior at low particle volume fractions, suggesting that in these cases HLBM works without an ex-

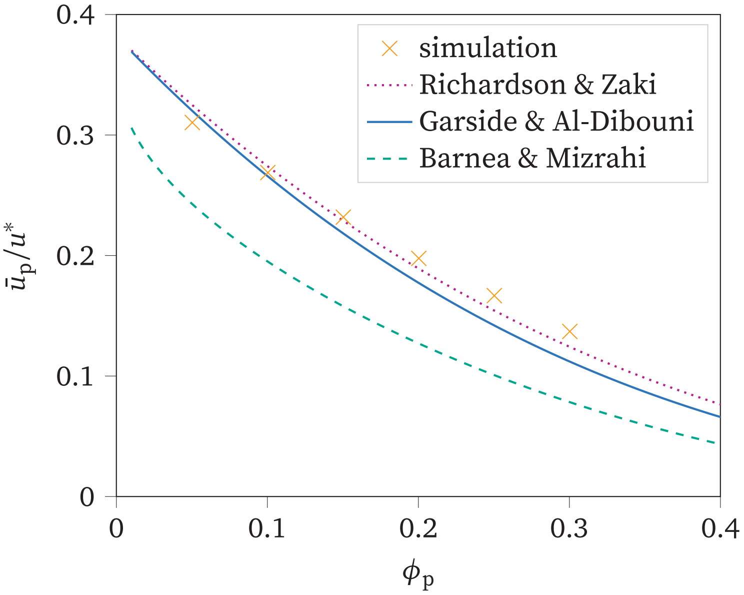
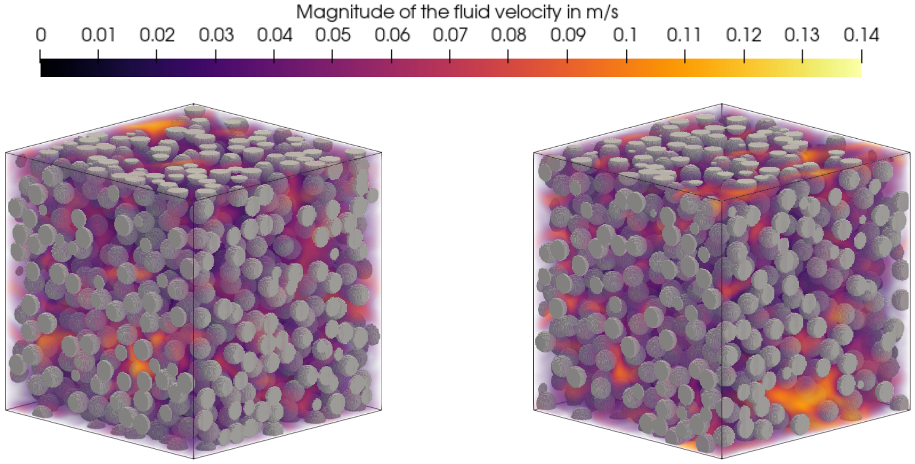


Figure 2.5: *Ratio of average settling velocity to single particle settling velocity $\bar{u}_p/u^*$ versus the particle volume fraction ϕ_p, showing current simulation results and the correlations by Richardson and Zaki [77], Garside and Al-Dibouni [79], as well as Barnea and Mizrahi [78].*

plicit contact model, since the interaction through the fluid is prominent. The observed discrepancy between the simulation results and the correlations at higher particle volume fractions can be attributed to the absence of an explicit contact model, because with increasing particle volume fraction, the probability of particle-particle interactions also increases. This in accordance with previous findings [69]. In simulations without an explicit contact model, these interactions are not fully accounted for, leading to an error in the predicted settling velocities. Since the energy dissipation due to interparticle interactions is missing, the observed overestimation of the velocity seems reasonable. The inclusion of a contact model [71, 89] would allow for a more accurate representation of particle-particle interactions, potentially reducing the difference between simulation results and correlations, especially at higher particle volume fractions. However, since we are primarily interested in quantifying the performance improvement from decomposition, an elaborate contact model would introduce too much bias by degrading overall performance by introducing more complex computation and communication.



(a) $t^* \approx 49.97$

(b) $t^* \approx 299.82$

Figure 2.6: *Particles and fluid velocity at different normalized times t^* for Ar $= 1500$ and $\phi_p \approx 0.3$.*

Figure 2.6 visualizes the fluid velocity and the particles for $\phi_\mathrm{p} \approx 0.3$ at different normalized times t^*. It can be seen that the particles form clusters and that high fluid velocity channels form. The observation is quantitatively consistent with the literature [55], further confirming the correctness. However, interparticle contacts are also visible, hinting at the potential need for an explicit contact model at high particle volume fractions.

2.5.4 Performance

In this section, we evaluate the performance of HLBM with the novel particle decomposition scheme and the conventional one. The computational infrastructure used for these experiments consists of Intel Xeon Platinum 8368 CPUs, with each node equipped with 76 CPU cores.

Figure 2.7 presents the results of this analysis. Here, we plot the million lattice site updates per second (MLUPs) versus the number of nodes used. We consider a fixed total problem size with a resolution of 12 in Figures 2.7(a) and 2.7(b), a resolution of 18 in Figures 2.7(c) and 2.7(d), and a resolution of 27 in Figures 2.7(e) and 2.7(f). In each scenario, the number of particles ranges from 166 to 991.

The performance plots in Figure 2.7 reveal several important observations: Firstly, the novel method exhibits in general higher MLUPs compared to the

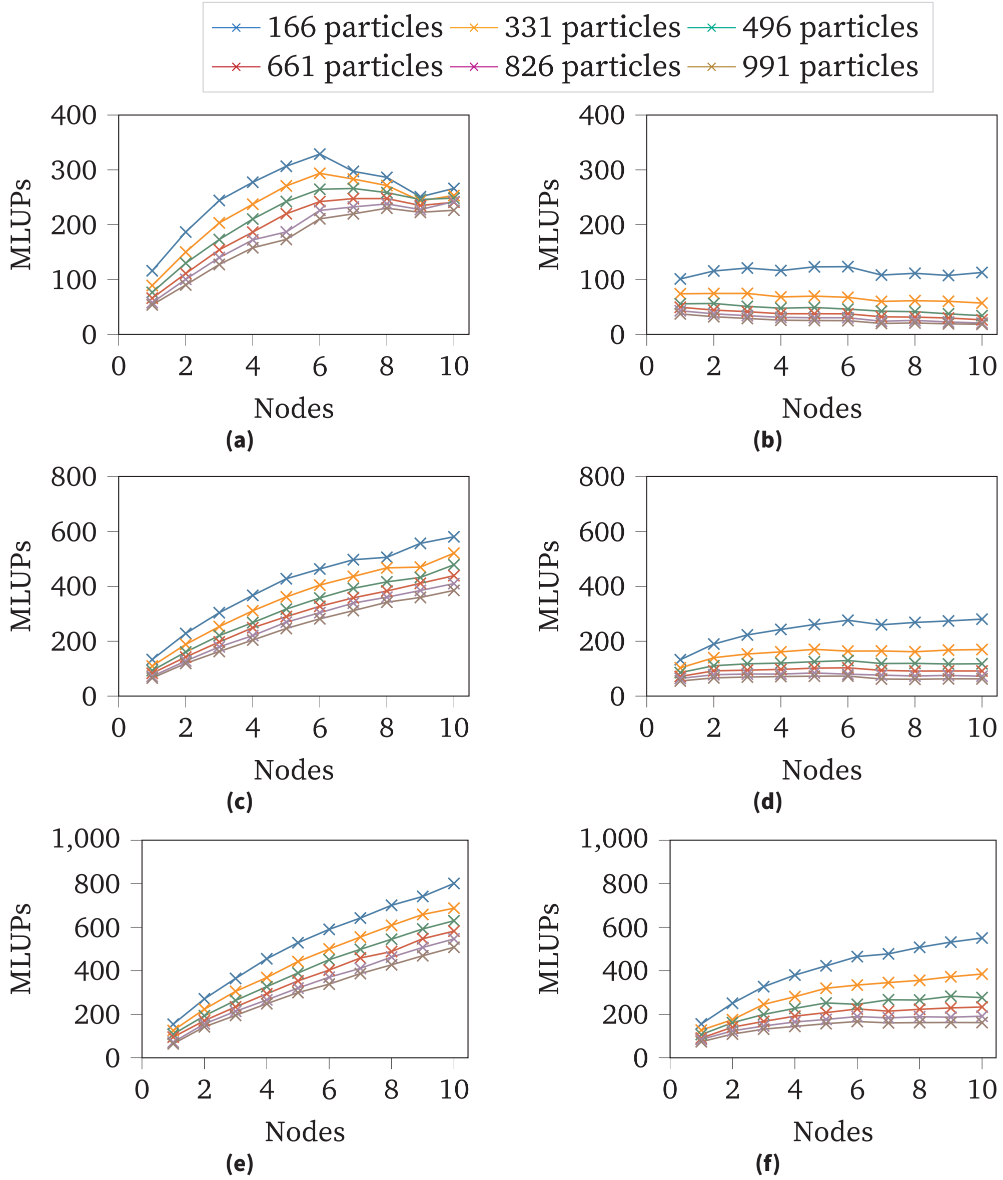


Figure 2.7: *Comparison of the resulting MLUPs versus the number of employed nodes using the new particle decomposition scheme (left) and the conventional decomposition (right) for the resolutions $N = 12$ (**a** and **b**), $N = 18$ (**c** and **d**) and $N = 27$ (**e** and **f**) and different particle numbers.*

conventional method. However, for small resolution and low particle number, the MLUPs are of a similar scale, but as particle number and resolution increase, the differences increase. For example, the simulations with the new particle decomposition show MLUPs more than five times larger than those with the conventional method, when considering 991 particles. Additionally, the conventional method demonstrates significant communication limitations for $N = 12$, which is immediately apparent by a flattening to a plateau in the plots, and for $N = 18$, which become noticeable when the number of nodes reaches six.

In contrast, the new method shows significant communication limitations only for $N = 12$ and when the number of nodes reaches seven or more, see Figure 2.7(a). Note that the problem size is smallest in this particular scenario. Increasing the number of processes beyond the observed inflection point degrades performance because communication becomes the limiting factor.

In a broader context, increasing the number of processes increases the number of costly communications, which explains the flattening of all performance curves as more nodes are used. At some point, communications become more expensive than computations, leading to the inflection point observed above. This behavior would occur in all cases, but shows up earlier for small problem sizes because the overall workload is smaller.

The new method demonstrates improved computational efficiency and scalability. These improvements are due to the more efficient communication strategy of the proposed method. Unlike the conventional method, which transmits all data to every process, the novel method selectively communicates data within the neighborhood of each block, as explained earlier. By minimizing unnecessary data transfer and concentrating communication on relevant blocks, the novel method significantly reduces communication overhead and optimizes computational resources.

These results highlight the benefits of the novel method in mitigating communication bottlenecks.

The studies focus on spheres, but a performance gain is expected for any other shape, because in these cases the signed distance function is more complex. The increased complexity means that the computations performed on each lattice node, see Section 2.3.2, are more expensive. It is therefore even more important to distribute the computational effort across the processes in cases of complex shapes. The relative performance gain is expected to be even higher.

The studies furthermore consider a minimum of 166 particles. However, the performance improvements are expected to also extend to single particle simulations, due to a reduction in communication overhead, although to a lesser extent. This reduction is due to the fact that only the processes adjacent to the singular particle are required to communicate, allowing the remaining processes to perform other computations in parallel.

2.6 Summary and Conclusions

In the present work, we propose and validate a novel and improved particle decomposition scheme for surface resolved particle simulations and apply it to HLBM for demonstration. For this purpose, we perform extensive studies using the example of hindered settling.

The main objective, performance improvement, is evaluated and confirmed, as the comparison of the conventional and the novel methods in terms of computational efficiency and communication constraints reveals significant advantages offered by the novel method. The novel method scales considerably better, resulting in higher MLUPs compared to the conventional method. This improvement is due to the more efficient communication strategy of the novel method, where data is selectively communicated.

The studies also show that HLBM is capable of capturing the settling behavior of a particle swarm at low particle volume fractions. This suggests that under these conditions, HLBM works effectively without the need for a dedicated contact model. However, as particle volume fractions increase, the limitations of not having a dedicated contact model become more pronounced. Therefore, at higher particle volume fractions, there is a clear need to incorporate a dedicated contact model to accurately simulate and account for particle interactions.

These results emphasize the importance of efficient communication strategies in particle simulation methods and highlight the value of the novel method in achieving computational efficiency. It is now feasible to simulate a larger number of surface resolved particles, resulting in more realistic simulation setups and expanded applicability to real-world problems, such as filters and thickeners.

3
Discrete Contact Model

This chapter has been published as

Marquardt, J. E. et al. "A discrete contact model for complex arbitrary-shaped convex geometries." In: Particuology *80 (2023), pp. 180–191.* ISSN*: 1674-2001.* DOI*: 10.1016/j.partic.2022.12.005*

3.1 Introduction

Particle-laden flows are essential in many fields such as solid-liquid separation and food processing. In various applications, a high particle concentration is present and, therefore, modeling requires four-way coupling. Thus, besides coupling of the fluid to the particle (one-way coupling) and particle to the fluid (two-way coupling), particle-particle and particle-wall interactions are also essential for a correct investigation. In food processing, for instance, the accurate calculation of acting forces is indispensable to avoid damage to the particulate phase, as this would reduce the quality of the end product.

An option to consider individual particles is the use of the discrete element method (DEM), which has many applications [33]. Resolving an arbitrary geometry is, e.g., possible by a simple approach of glued spheres [41]. In a similar fashion, the framework Grains3D [43, 45] enables modeling the contact of non-convex particle geometries through a description of glued convex shapes [44]. Yet, the arbitrary shapes' approximation by, e.g., spheres leads to a limited accuracy. To increase the accuracy, the number of sphere segments must also increase immensely, leading to expensive computations. Additionally, to model the influence of a surrounding fluid remains a challenge as all the above mentioned models consider dry collisions. Although, studies regarding the coupling with a surrounding fluid exist. However, these mostly use simple spherical geometries [46, 47] or solely model the drag coefficient and lack a back coupling from the particles to the fluid [48]. Thus, to process an accurate four-way coupling with realistic and complex shapes, resolving these by direct numerical simulations (DNS) is necessary.

Nagata et al. [59] proposed an immersed boundary method (IBM) collision algorithm that is suitable for arbitrary shapes and shows good accuracy for simple geometries. The underlying IBM resolves surfaces using Lagrange points [58]. These points interact with the fluid, but need not depend on the particular fluid grid. As a result, IBM achieves a high accuracy. However, the frequent interpolations between particle and fluid points are costly. Also, it is beneficial that it is coupleable with different approaches for solving the fluid, such as the finite element method, the finite volume method, and the lattice Boltzmann method (LBM).

The latter, LBM, is of increasing interest as it is efficient and is easy to parallelize because costly computations are purely local [65]. Other LBM-based approaches to simulate arbitrary particle shapes also exist. The partially saturated cells method (PSM) is the most common and was first proposed by Noble and Torczynski [61] in its original form. Since then, there have been several new approaches, such as the homogenized lattice Boltzmann method (HLBM) introduced by Krause et al [66]. All PSMs use a level-set function to describe an approximation of the volume fraction of the particle over the complete computation domain [62]. Previous studies using HLBM show that it allows representation of almost every shape [67–69]. Since the method uses the LBM voxel representation, the approximation of the shape, depends on the grid, and very thin objects are prone to problems. Though PSMs have shown potential, modeling the contact of arbitrary-shaped particles, which is essential to consider realistic suspensions with high particle concentrations, remains a big challenge. Although previous studies show that lubrication forces suffice to decelerate particles, the influence of contacts is still incorrectly represented [69].

The aim of this paper is to provide a model to consider the contact of complex convex particle shapes to close the above gap. The literature lacks viable options to consider high particle volume fractions along with fluid interaction, when the particle geometry is realistic and complex. For this purpose, we propose a discretized contact model that is usable along with PSMs, such as HLBM. In other words, this paper introduces a novel model for dense suspensions using PSMs and validates it on suitable test cases. The associated simulations for validation use OpenLB [94, 102].

To introduce the method and its applicability, the paper adopts the following structure. In Section 3.2, the models used to consider the fluid, particles, and interactions are introduced. In Section 3.3, we discuss the numerical methods that are used to solve the model system. This is followed by a validation of the novel method for contact treatment and its application in a system with and without a particle-laden viscous fluid in Sections 3.4 and 3.5. Finally, in Section 3.6, we give a summary and draw a conclusion.

3.2 Modeling

Suspensions consist of fluids and particles. Both components need to be described by a proper model. This also applies to the particle-particle and particle-wall interaction in the presence of a high particle concentration. For these interactions, the geometries of the objects under consideration are of great importance. Therefore, we show the models for fluids in Section 3.2.1, for particles in Section 3.2.2, for geometries in Section 3.2.3, and for contacts in Section 3.2.4.

3.2.1 Fluid

We describe the fluid component by the incompressible Navier–Stokes equations:

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\,\boldsymbol{u} - \frac{\eta}{\rho}\Delta \boldsymbol{u} + \frac{1}{\rho}\nabla p &= \boldsymbol{F}_{\mathrm{f}}, \\ \nabla \cdot \boldsymbol{u} &= 0, \end{aligned} \tag{3.1}$$

where $\boldsymbol{u}$ is the fluid velocity, t the time, p the pressure, $\boldsymbol{F}_{\mathrm{f}}$ the total of all forces acting on the fluid, e.g., gravity, η and ρ are the fluid's dynamic viscosity and density.

3.2.2 Particle

To consider particles, we need to describe the particle motion and represent the particle's shape. For the former, we use Newton's second law of motion. The translation is then given by

$$m\frac{\partial \boldsymbol{v}_\mathrm{p}}{\partial t} = \boldsymbol{F}_\mathrm{p} \tag{3.2}$$

and the rotational motion by

$$\boldsymbol{I}_\mathrm{p}\frac{\partial \boldsymbol{\omega}}{\partial t} = \boldsymbol{T}_\mathrm{p}. \tag{3.3}$$

In the above-mentioned equations, m, $\boldsymbol{I}_\mathrm{p}$, $\boldsymbol{v}_\mathrm{p}$, $\boldsymbol{\omega}$, $\boldsymbol{F}_\mathrm{p}$, $\boldsymbol{T}_\mathrm{p}$ are the particle's mass, moment of inertia, velocity, angular velocity, and the sum of forces as well as torque acting on the particle. Here, all quantities with index p refer to the particle's center of mass. Note that in this work $\boldsymbol{F}_\mathrm{p}$ is composed of hydrodynamic forces $\boldsymbol{F}_\mathrm{p,h}$, normal and tangential contact forces $\boldsymbol{F}_\mathrm{n}$ and $\boldsymbol{F}_\mathrm{t}$. The torque acting on the particle $\boldsymbol{T}_\mathrm{p}$ also stems from the hydrodynamic and the contact forces. For the latter, the lever results from the difference between the contact point $\boldsymbol{x}_\mathrm{c}$ and the center of mass of the particle $\boldsymbol{x}_\mathrm{m}$.

3.2.3 Geometries

In this work, we use signed distance functions (SDFs) to describe the geometries of the particles and walls. Since a comprehensive discussion of these would exceed the scope of this paper, the interested reader is referred to the literature [103, 104] for more information.

We use SDFs to determine the signed distance d_s of a point $\boldsymbol{x}$ to a surface $B \in \mathbb{R}^D$ with the dimension D. Here, the sign indicates whether the point $\boldsymbol{x}$ is inside or outside the geometry. This distance to the surface is particularly valuable for contact modeling and allows us to derive a significant part of the necessary quantities. For example, the normal on the geometry surface is obtained from the derivative of the SDF. Stemming from computer graphics, SDFs are also highly efficient. These advantages make them a perfect fit for the application discussed here.

Henceforth, we use

$$d_\mathrm{s}(\tilde{\boldsymbol{x}}) = ||\tilde{\boldsymbol{x}}|| - R \tag{3.4}$$

to describe spheres and

$$d_\mathrm{s}(\tilde{\boldsymbol{x}}) = \max\left(||(\tilde{x}_x, \tilde{x}_y)|| - R, \tilde{x}_z - H/2\right) \tag{3.5}$$

to represent cylinders which have an axis orthogonal to the x-y-plane. Here, R is the radius, H the cylinder's length and $\tilde{\boldsymbol{x}}$ is location relative to the center of mass of the respective geometry. The transformation from the global coordinate system of the simulation into the coordinate system of the geometry uses translation and rotation depending on the motion of the particle.

3.2.4 Contact

In order to describe interactions of arbitrary geometries, we use a model proposed by Nassauer and Kuna [105] to compute the normal, Section 3.2.4.1, and tangential, Section 3.2.4.2, contact force from an overlap volume. The commonly very small overlap models deformation during the contact and does not occur in reality.

3.2.4.1 Normal contact

Following this model, the normal contact force's magnitude reads

$$F_\mathrm{n} = E^* k \sqrt{V_\mathrm{c} d}\left(1 + c\dot{d}_\mathrm{n}\right), \tag{3.6}$$

with the effective modulus of elasticity E^*, a constant k, the overlap volume V_c, the indentation depth d, a damping constant c, and the magnitude of the relative velocity between two bodies in contact in the direction of the normal force $\dot{d}_\mathrm{n}$. The former is given as

$$E^* = \left(\frac{1-\nu_\mathrm{A}^2}{E_\mathrm{A}} + \frac{1-\nu_\mathrm{B}^2}{E_\mathrm{B}}\right)^{-1}. \tag{3.7}$$

Here, E_A and ν_A are the modulus of elasticity and the Poisson's ratio of objects A and E_B and ν_B of object B, which are in contact.

The constant $k = 4/(3\sqrt{\pi})$ applies to a sphere–half-space and sphere–sphere contact [105]. For a cylindrical flat punch, on the other hand, $k = 2/\sqrt{\pi}$ applies.

By multiplication with the contact normal $\boldsymbol{n}_\mathrm{c}$, we obtain vector components from the magnitude of the force

$$\boldsymbol{F}_\mathrm{n} = \boldsymbol{n}_\mathrm{c} F_\mathrm{n}, \tag{3.8}$$

which is necessary for solving the particle motion equations, see Eqs. (3.2) to (3.3).

To solve the above-mentioned model equations, we need to determine the overlap volume V, indentation depth d, and contact normal $\boldsymbol{n}_\mathrm{c}$ numerically, as presented in Section 3.3.3.

Also, a correct damping factor, which depends on the initial relative velocity in the normal contact direction, is crucial. However, the consideration of this correlation is beyond the scope of this paper. Nevertheless, it is vital to evaluate the applicability of existing models from the literature [106, 107] in future works.

3.2.4.2 Tangential contact

Tangential forces occur due to friction. Commonly, friction depends on the normal force , see Eq. (3.6), and the coefficients of static and kinetic/sliding friction, μ_s and μ_k, respectively. Nassauer and Kuna [105] follow a similar logic and describe the magnitude of the friction force by

$$F_\mathrm{t} = \left(\left(2\mu_\mathrm{s}^* - \mu_\mathrm{k} \right) \frac{a^2}{a^4 + 1} + \mu_\mathrm{k} - \frac{\mu_\mathrm{k}}{a^2 + 1} \right) F_\mathrm{n}, \tag{3.9}$$

with

$$\mu_\mathrm{s}^* = \mu_\mathrm{s} \left(1 - 0.09 \left(\frac{\mu_\mathrm{k}}{\mu_\mathrm{s}} \right)^4 \right), \tag{3.10}$$

and

$$a = \frac{||\boldsymbol{v}_{\mathrm{AB,t}}(\boldsymbol{x}_\mathrm{c})||}{v_s}. \tag{3.11}$$

Here, we use the relative tangential velocity $\boldsymbol{v}_{\mathrm{AB,t}}$ at the contact point $\boldsymbol{x}_\mathrm{c}$ of the objects A and B that are in contact and the model parameter v_s, which denotes the velocity at which a transition from static to kinetic friction occurs. The friction force acts against the relative tangential velocity $\boldsymbol{v}_{\mathrm{AB,t}}$ and thus

$$\boldsymbol{F}_\mathrm{t} = -F_\mathrm{t} \frac{\boldsymbol{v}_{\mathrm{AB,t}}(\boldsymbol{x}_\mathrm{c})}{||\boldsymbol{v}_{\mathrm{AB,t}}(\boldsymbol{x}_\mathrm{c})||} \tag{3.12}$$

applies.

3.3 Numerical methods

3.3.1 Lattice Boltzmann method

To solve the incompressible Navier–Stokes equations mentioned in Section 3.2.1, we use the lattice Boltzmann method (LBM) [65, 90, 91].

The origin of LBM is gas kinetics, and for this reason, it operates on a mesoscopic scale and considers particle populations. We describe these populations by a discrete velocity distribution function $f_i(\boldsymbol{x}, t)$. The arguments of the particle populations, the position $\boldsymbol{x}$ and the time t, are discrete. Also, there is a discrete set of velocities $\boldsymbol{c}_i$, which is predetermined. For example, in the following, we use a D3Q19 velocity set for all of our simulations, which operates in a three-dimensional space and has 19 discrete velocities [65, 90]. We can additionally use the distribution function to calculate macroscopic quantities, such as density $\rho(\boldsymbol{x},t) = \sum_i f_i(\boldsymbol{x},t)$ and velocity $\rho\boldsymbol{u}(\boldsymbol{x},t) = \sum_i \boldsymbol{c}_i f_i(\boldsymbol{x},t)$.

To compute the distribution function's value in the next time step, we use the lattice Boltzmann equation (LBE) with an additional source term $S_i(\boldsymbol{x},t)$ to account for forces acting on the fluid $\boldsymbol{F}_\mathrm{f}$, which in this context is the influence of submerged particles via the HLBM, see Section 3.3.2. The LBE is usually utilized with a time step size $\Delta t = 1$ in lattice units and divided into two distinct parts. The collision step

$$f_i^*(\boldsymbol{x},t) = f_i(\boldsymbol{x},t) + \Omega_i(\boldsymbol{x},t) + S_i(\boldsymbol{x},t), \tag{3.13}$$

with the collision operator Ω_i, through which we obtain the post-collision distributions f_i^*, and the streaming step

$$f_i(\boldsymbol{x}+\boldsymbol{c}_i, t+1) = f_i^*(\boldsymbol{x},t), \tag{3.14}$$

which distributes the post-collision distribution f_i^* to the neighboring lattice nodes. Furthermore, we use the Bhatnagar–Gross–Krook (BGK) collision operator [92], which is given by

$$\Omega_i(\boldsymbol{x},t) = -\frac{f_i(\boldsymbol{x},t) - f_i^{\mathrm{eq}}(\rho,\boldsymbol{u})}{\tau}, \tag{3.15}$$

where τ is the relaxation time that governs the speed of the particle populations' relaxation towards its equilibrium state. The equilibrium is given by the discrete Maxwell–Boltzmann distribution

$$f_i^{\mathrm{eq}}(\rho,\boldsymbol{u}) = w_i\rho\left(1 + \frac{\boldsymbol{c}_i\cdot\boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{c}_i\cdot\boldsymbol{u})^2}{2c_s^4} + \frac{\boldsymbol{u}^2}{2c_s^2}\right). \tag{3.16}$$

The necessary weights w_i and constant lattice speed of sound c_s depend on the chosen velocity set. The former comes from a Gauss–Hermite quadrature rule, and $c_s = 1/\sqrt{3}$ when using a D3Q19 set.

3.3.2 Homogenized lattice Boltzmann method

Trunk et al. [69] distinguish between three different elements in the method. Namely, object representation, forcing scheme, and momentum exchange.

The former uses a voxel representation, which was explained in detail by Trunk et al. [67]. This representation allows to calculate necessary physical properties such as volume, mass, center of mass, and moment of inertia. Additionally, we map the particle onto the fluid domain during this step. For this, we use a model parameter $d_B \in [0,1]$ to calculate the local velocity difference

$$\Delta \boldsymbol{u}(\boldsymbol{x},t) = d_B(\boldsymbol{x},t)(\boldsymbol{v}(\boldsymbol{x},t) - \boldsymbol{u}(\boldsymbol{x},t)). \tag{3.17}$$

Here, $\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{v}(\boldsymbol{x}_m) + \boldsymbol{\omega} \times (\boldsymbol{x} - \boldsymbol{x}_m)$ is the particle's velocity at position $\boldsymbol{x}$, and $\boldsymbol{x}_m$ is the particle's center of mass. The aforementioned smoothing at the particle surface uses trigonometric functions, as described by Krause et al. [66], with the signed distance to the particle surface. We use

$$d_B(\boldsymbol{x},t) = x = \begin{cases} 1 & \text{if } d_\mathrm{s} \leq -\varepsilon/2 \\ \cos^2(\pi\left(\frac{d_\mathrm{s}}{\varepsilon} + \frac{1}{2}\right) & \text{if } \varepsilon/2 > d_\mathrm{s} > -\varepsilon/2 \\ 0 & \text{if } d_\mathrm{s} \geq \varepsilon/2 \end{cases} \tag{3.18}$$

to compute the model parameter and set the size of the smooth boundary via the parameter ε.

Since studies by Trunk et al. [69] show that it provides the best results, we use the adaption of the exact difference method (EDM) by Kupershtokh et al. [95] as the forcing scheme as well. The method solely uses the source term

$$S_i(\boldsymbol{x},t) = f_i^{\mathrm{eq}}(\rho, \boldsymbol{u} + \Delta\boldsymbol{u}) - f_i^{\mathrm{eq}}(\rho, \boldsymbol{u}) \tag{3.19}$$

in Eq. (3.13) with the above-mentioned local velocity difference $\Delta\boldsymbol{u}$, see Eq. (3.17).

To obtain the hydrodynamic force acting on the particle $\boldsymbol{F}_{\mathrm{p,h}}$ and the resulting torque $\boldsymbol{T}_{\mathrm{p,h}}$, we use the momentum exchange algorithm (MEA) by Wen et al. [96]. We then compute the acceleration from the resulting hydrodynamic force along with other forces, such as gravitation and buoyancy, via Newton's second law of motion and use it within an explicit time integration scheme. Here, we choose the velocity Verlet algorithm [108, 109] to solve the equations of motion.

3.3.3 Discrete contacts

In the following, we propose a discrete method to treat particle-particle and particle-wall interactions. To do so, we need to find a simple cuboid with which

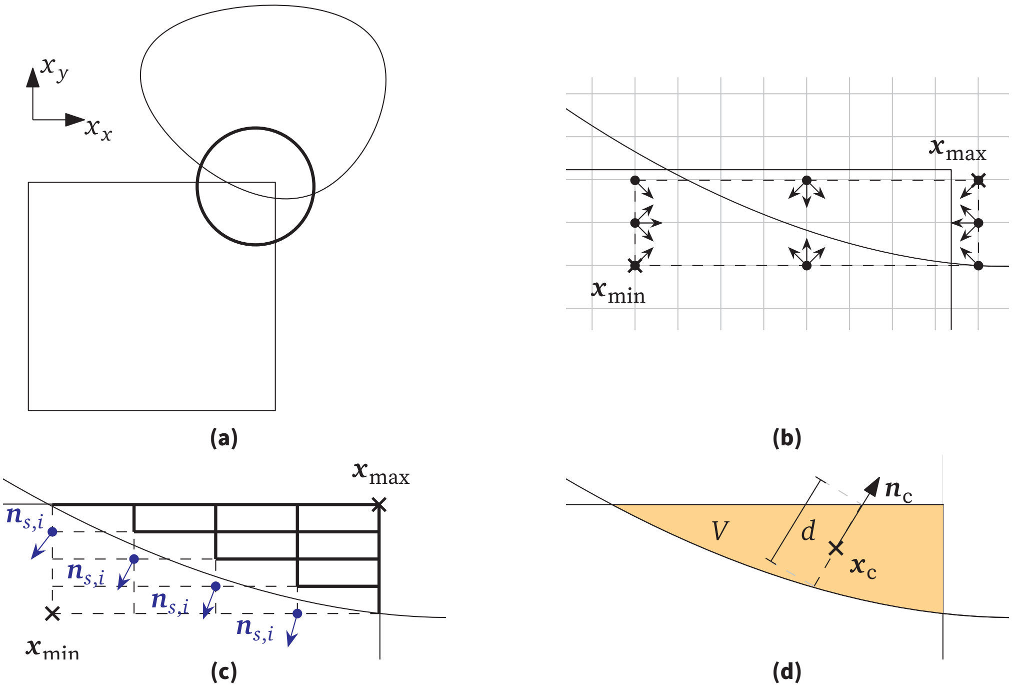


Figure 3.1: *Illustration of two convex objects in contact. Showing a rectangle and a rounded geometry and highlighting the overlap area with a thickened circle (a). (b) shows the magnified contact area with an example lattice in gray and the approximated bounding box. It also presents the magnified contact with the improved bounding box and the grid used for the calculation of all contact-relevant properties (c). Furthermore, the contact and the necessary properties are displayed in (d).*

the contact geometry is enclosed, see Section 3.3.3.1. In the following, we refer to this enclosing cuboid as bounding box. Afterwards, we must correct this bounding box as presented in Section 3.3.3.2. We then iterate over the improved bounding box with discrete steps to calculate the contact properties in Section 3.3.3.3. The necessary steps and their outcome are illustrated in Figure 3.1.

3.3.3.1 Contact detection

In Figure 3.1(a), we see an exemplary contact. For simplification, this is a 2D contact, however, the following analogously applies to 3D. It is easy to see that the two objects are in contact. However, this is much more difficult to determine numerically for complex geometries.

To overcome this challenge, we use a very simple detection while setting the HLBM model parameter d_B on the lattice, since in both cases, we have a dependency on the signed distance d_s. For clarity, Figure 3.1(b) shows the magnified contact area with the lattice drawn in gray. Each grid point where the signed distance to both object surfaces is less than half of a cell's diagonal, $d_\mathrm{s} < 0.5\sqrt{D}\Delta x$, with the considered dimension D, we consider being inside the overlap. If a point is inside the contact according to the previously mentioned condition, we update the minimum and maximum coordinates, $\boldsymbol{x}_\mathrm{min}$ and $\boldsymbol{x}_\mathrm{max}$, of the bounding box, represented by the dashed line. Because of this, Figure 3.1(b) illustrates an overestimation of the bounding box. This is advantageous, however, as it increases the accuracy and stability. Otherwise, a higher fraction of the overlap is incorrect and, for the next step presented in Section 3.3.3.2, an overestimation is better than an underestimation. Because underestimation can lead to erroneous missing contact detections that are impossible to correct with currently known approaches and reasonable effort.

The contacts found are saved in contact objects. For example, we store the particle IDs and the minimum as well as maximum coordinates, $\boldsymbol{x}_\mathrm{min}$ and $\boldsymbol{x}_\mathrm{max}$, of the bounding box of the contact for particle-particle contacts in it. For particle-wall interactions, we solely need to replace one particle ID with an identifier for the wall. This means that all data is available for the later correction of the bounding box (Section 3.3.3.2) or contact force calculation (Section 3.3.3.3).

The aforementioned identifiers are, e.g., simple indices of a field which stores the particles or walls. Naturally, other implementations are also plausible. It is only important that this identifier is unique and constant during the contact treatment, or, if a change is absolutely necessary, then the changes must also be applied to the contact object.

3.3.3.2 Correct detected contacts

Since the rough contact detection most likely overestimates the bounding box, we need to shrink it to the actual contact , using Algorithm 3.1. This step causes the change from Figure 3.1(b) to (c). Thereby, we improve the accuracy of the contact force calculation, which we discuss in Section 3.3.3.3. To do this, we

Algorithm 3.1: Algorithm to correct the initial bounding box.

```
Calculate step size for rough bounding box ;                ▷ See Eq. (3.20)
for all discrete positions inside the bounding box do
    if position is on bounding box surface then
        // Iterate over all discrete directions as illustrated
           in Figure 3.1(b)
        for all discrete directions in 45° steps do
            if neighbor is inside the bounding box then
                // Calculate distance via ray marching or ray
                   tracing
                Calculate distance to the collision surface;
                Determine position on the collision surface;
                Update the minimal and maximal bounding box coordinates;
            end
        end
    end
    collision & streaming;
end
```

first calculate a step size

$$\Delta x_{c,j} = \frac{1}{N_c} \begin{cases} \Delta x & \text{if } x_{\max,j} - x_{\min,j} = 0, \\ (x_{\max,j} - x_{\min,j}) & \text{if } x_{\max,j} - x_{\min,j} > 0. \end{cases} \tag{3.20}$$

with which we iterate over the overlap volume's bounding box per spatial direction j. This resolves the overlapping area with N_c cuboids, which may have different aspect ratios.

In the following, we iterate over the bounding box's surface, using the step sizes from Eq. (3.20). During this iteration, we determine the distances to the real contact surface, which is inside the bounding box, in different discrete directions. These directions change in 45° steps as illustrated for selected grid points (black disks) in Figure 3.1(b). Based on the point on the surface, the direction and the distance determined, we can now identify points on the contact surface and derive an improved bounding box. The distance to the contact surface can be calculated, e.g., via ray marching or ray tracing.

In the end, we thus obtain a bounding box of the overlapping region of two identifiable objects. This gives us sufficient information to calculate the contact forces in the next step.

3.3.3.3 Calculation of contact forces

Having obtained the bounding box, we continue with the calculation of the forces acting on the objects in contact. For this, we apply another uniform rectangular grid on the bounding box, as shown in Figure 3.1(c), to obtain the necessary parameters for the calculation of the contact force. Some of them are shown in Figure 3.1(d). We again compute the distance between the respective grid points from Eq. (3.20). To derive the contact information, as mentioned in Section 3.2.4, we evaluate if it lies within the contact (solid lines) or outside the contact (dashed lines), on each grid point, and count the total number of cells n_{cell} within the overlap.

Overlap volume. We then calculate the overlap volume V_{c} of two colliding objects by a sum of the volume of all individual cells that are within the contact:

$$V_{\mathrm{c}} = n_{\mathrm{cell}} V_{\mathrm{cell}}. \tag{3.21}$$

Here, V_{cell} is each cell's volume. In Figure 3.1(c), we see that a point may lie right on the surface of the contact, which therefore would lead to an overestimation of the overlap volume. However, this is solely a discretization error and decreases drastically with an increased resolution. Also, in actual simulations, such an ideal contact is almost impossible, so, if at all, the minimum and maximum are on the surface of the contact and all other grid points are inside. It is therefore advantageous to use intersections of the grid lines and not the centers as it compensates for the otherwise missing volume to an extent.

Contact normal. Another required quantity is the contact normal $\boldsymbol{n}_{\mathrm{c}}$. We calculate it from the cells on the surface, but we use the first layer of grid points outside the contact. In Figure 3.1(c), some considered points and their respective normals $\boldsymbol{n}_{\mathrm{s},i}$ are presented in blue. This leads to greater accuracy, because we have more cells in this layer than in the inner one for convex shapes. Additionally, we verify that a point is actually on the surface by checking that the neighbor in the direction of the normalized normal at the cell's center $\boldsymbol{n}_{\mathrm{s},i}$ is inside the overlap. The normal $\boldsymbol{n}_{\mathrm{s},i} = \nabla d_{\mathrm{s}}(\boldsymbol{x})$ is obtained from the derivative of the SDF at the corresponding position. To calculate this, we use the central difference method. Analogous to the procedure of Nassauer and Kuna [105], we calculate the normal from a weighted average of the normals of the previously defined surface cells

$$\boldsymbol{n}_{\mathrm{c}} = \frac{\sum_i^{n_{\mathrm{cell,s}}} A_i \frac{\boldsymbol{n}_{\mathrm{s},i}}{||\boldsymbol{n}_{\mathrm{s},i}||}}{\sum_i^{n_{\mathrm{cell,s}}} A_i}. \tag{3.22}$$

Here, we sum over all surface cells $n_{\text{cell,s}}$ and weigh each with the surface A_i, which is the cross section in the cell's normal direction $\boldsymbol{n}_{\text{s},i}$. It is further important to note that we use $\boldsymbol{n}_{\text{s},i} = -\nabla d_\text{s}(\boldsymbol{x})$ for the second object in contact, so that the normals of both objects point in the same direction. We estimate A_i in 3D with

$$A_i = \begin{pmatrix} \Delta x_{c,y} \Delta x_{c,z} \\ \Delta x_{c,x} \Delta x_{c,z} \\ \Delta x_{c,x} \Delta x_{c,y} \end{pmatrix} \cdot \left(\frac{\boldsymbol{n}_{\text{s},i}}{||\boldsymbol{n}_{\text{s},i}||} \odot \frac{\boldsymbol{n}_{\text{s},i}}{||\boldsymbol{n}_{\text{s},i}||} \right), \tag{3.23}$$

and in 2D with

$$A_i = \begin{pmatrix} \Delta x_{c,y} \\ \Delta x_{c,x} \end{pmatrix} \cdot \left(\frac{\boldsymbol{n}_{\text{s},i}}{||\boldsymbol{n}_{\text{s},i}||} \odot \frac{\boldsymbol{n}_{\text{s},i}}{||\boldsymbol{n}_{\text{s},i}||} \right), \tag{3.24}$$

In the equations above, ⊙ refers to the Hadamard product, i. e., an element-wise product.

Contact point. According to Nassauer and Kuna [105], the contact point $\boldsymbol{x}_c$ is defined as the center of mass of the overlap volume. This can be determined by dividing this area into cuboids, as shown in Figure 3.1(c), and by assuming that the particles have a constant density distribution. Then, we simply calculate the mean value of all cell centroids $\boldsymbol{X}_i$

$$\boldsymbol{x}_c = \frac{\sum_i^{n_\text{cell}} \boldsymbol{X}_i}{n_\text{cell}}. \tag{3.25}$$

Displacement. Now, the displacement d is to be determined. For this purpose, the distance from the contact point $\boldsymbol{x}_c$ to the contact surface is determined in two directions. On the one hand, in the direction of the contact's normal $\boldsymbol{n}_\text{c}$ and, on the other hand, in the opposite direction $-\boldsymbol{n}_\text{c}$. Both distances together result in the wanted displacement d, as illustrated in Figure 3.1(d). The temporal change of the displacement d reads [105]

$$\dot{d}_\text{n} = ||\boldsymbol{v}_{\text{AB,n}}(\boldsymbol{x}_c)||. \tag{3.26}$$

Here, we use the relative velocity in normal direction [110]

$$\boldsymbol{v}_{\text{AB,n}}(\boldsymbol{x}_c) = \left(\boldsymbol{v}_\text{AB}(\boldsymbol{x}_c) \cdot \frac{\boldsymbol{n}_\text{c}}{||\boldsymbol{n}_\text{c}||} \right) \frac{\boldsymbol{n}_\text{c}}{||\boldsymbol{n}_\text{c}||}, \tag{3.27}$$

at the contact point $\boldsymbol{x}_c$. The total relative velocity of two objects in contact, A and B, is calculated via

$$\boldsymbol{v}_\text{AB}(\boldsymbol{x}_c) = \boldsymbol{v}_\text{A}(\boldsymbol{x}_c) - \boldsymbol{v}_\text{B}(\boldsymbol{x}_c). \tag{3.28}$$

Algorithm 3.2: Basic LBM time step algorithm with four-way coupling.

```
for all time steps do
    Couple fluid to particles;                                  ▷ Using the MEA
    Couple particles to fluid with simultaneous, rough contact detection;
     ▷ Using the EDM and Section 3.3.3.1
    Communicate detected collisions;
    for all sub time steps do
        Correct bounding box;                         ▷ See Section 3.3.3.2
        Determine contact properties;                 ▷ See Section 3.3.3.3
        Calculate contact force;                        ▷ See Section 3.2.4
        Add external forces;                                ▷ E.g. gravity
        Solve equations of motion;
    end
    Delete empty collision objects;
    Perform collision and streaming;
    Increase time step;
end
```

Tangential velocity. The tangential velocity is the difference between the total relative velocity $\boldsymbol{v}_{\mathrm{AB}}$ and its normal component $\boldsymbol{v}_{\mathrm{AB,n}}$ [110]:

$$\boldsymbol{v}_{\mathrm{AB,t}} = \boldsymbol{v}_{\mathrm{AB}} - \boldsymbol{v}_{\mathrm{AB,n}}. \tag{3.29}$$

3.3.4 Time step algorithm with four-way coupling

For a better overview, Algorithm 3.2 shows the basic LBM time step algorithm with the four-way coupling. Here, the previously explained methods are listed in the necessary order. Additionally, we show the possibility of sub-time steps solely for the solution of the equations of motion, with a time step size $\Delta t_{\mathrm{c}} = \Delta t / n_{\mathrm{t}}$, for $n_{\mathrm{t}} \in \mathbb{N}_{>0}$. Here, n_{t} is the number of sub-time steps to process.

3.4 Validation

In the subsequent section, we consider several cases for validation. First, in Section 3.4.1, we validate the normal contact force. This is followed by a cylinder-wall impact test to check the resulting particle motion in Section 3.4.2.

3.4.1 Contact force

As a first challenge for the proposed method, we consider several contact problems with known analytical solutions, as described in Section 3.4.1.1. This is followed by the numerical results and a comparison of these with the analytical solutions in Section 3.4.1.2. Finally, in Section 3.4.1.3, we perform a grid independence study.

3.4.1.1 Setup

To validate the correctness of the normal contact force, we consider multiple test cases as illustrated in Figure 3.2. The tests include: a contact between a sphere with radius R and a half-space, see Figure 3.2(a), a contact between two spheres with different radii R_1 and R_2, see Figure 3.2(b), a cylindrical flat punch, see Figure 3.2(c), and a contact between two perpendicular crossed cylinders with an equal radius R, see Figure 3.2(d).

In the aforementioned tests, we increase the indentation depth incrementally and compare the results with given analytical solutions. For a contact between a sphere with radius R and a half-space, we expect the normal force [111]

$$F_\mathrm{n} = \frac{4}{3} E^* R^{1/2} d^{3/2}. \tag{3.30}$$

The equation looks similar for a contact between two crossed cylinders with the same radius R and for a contact of two spheres with radii R_1 and R_2. However, for the latter we must replace the radius R with the effective radius R_r reading [112]

$$R_\mathrm{r} = \left(\frac{1}{R_1} + \frac{1}{R_2} \right)^{-1}. \tag{3.31}$$

According to Popov et al. [111], the resulting normal contact force of a contact between a rigid cylinder with radius R and an elastic half-space is given by

$$F_\mathrm{n} = 2 R E^* d. \tag{3.32}$$

For the following tests, we consider a contact of two materials with $E_\mathrm{A} = 0.01$ GPa, $\nu_\mathrm{A} = 0.5$ as well as $E_\mathrm{B} = 200$ GPa and $\nu_\mathrm{B} = 0.3$. Additionally, we resolve the smallest diameter with eight cells for the contact detection and use $k = 4/(3\sqrt{\pi})$ for all except the cylindrical flat punch case, where we use $k = 2/\sqrt{\pi}$.

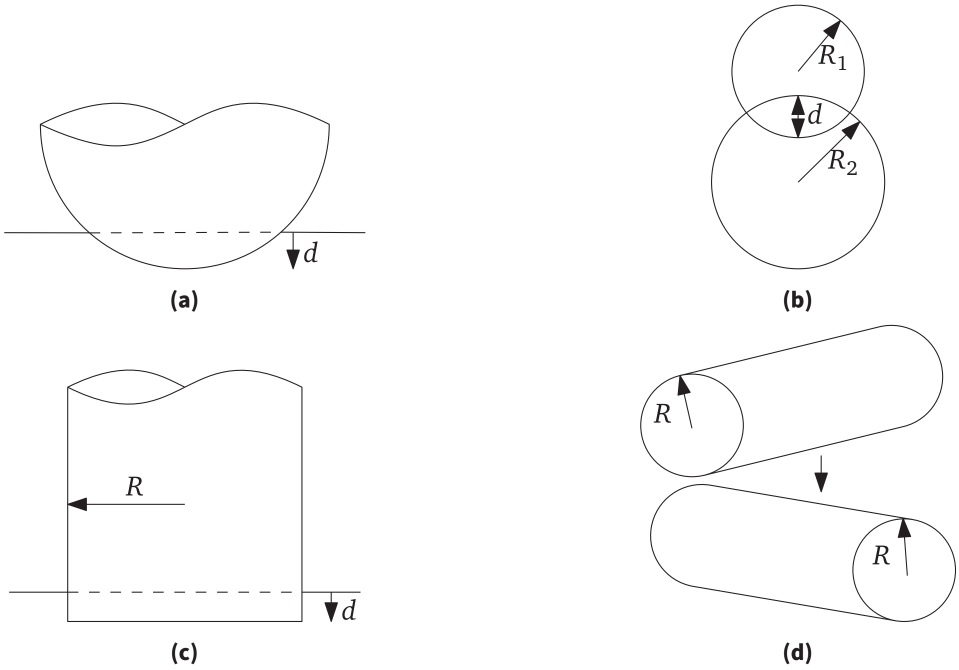


Figure 3.2: *Sketches of the considered test cases with an indentation depth d: (**a**) contact between a sphere with radius R and a half-space, (**b**) contact between two spheres with radii R_1 and R_2, (**c**) contact between a cylinder with radius R and a half space, and (**d**) contact between two perpendicular crossed cylinders with the same radius R.*

3.4.1.2 Results

Figure 3.3 shows the comparison of the analytical solutions as a solid line and computational results with different contact resolutions N_c. It is evident that for smaller resolutions, the resulting force has large deviations. Sometimes, the contact force is even mistakenly assumed to be 0. However, the results improve with increasing resolution, as expected. For a resolution of $N_c = 8$, this already leads to a good agreement of the simulation results with the analytical solutions, and larger resolutions provide an even better accuracy.

3.4.1.3 Grid independence

To confirm the grid independence, we calculate the root-mean-square relative error from the numerical results and analytical solutions provided in Section 3.4.1.2. Previously, we considered different radii. In Figure 3.4, however,

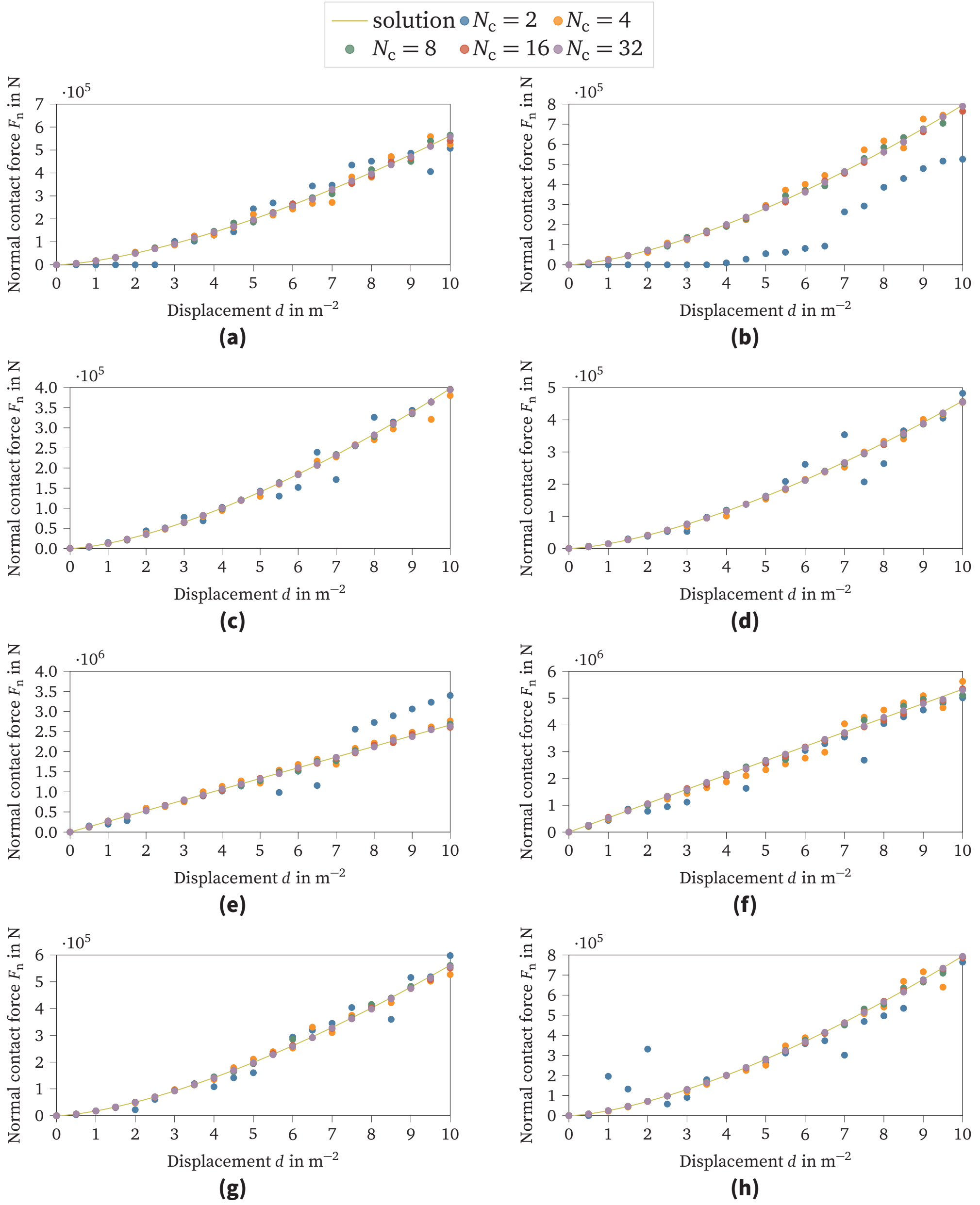


Figure 3.3: *Comparison of the analytical solution for the normal contact force F_n over the indentation depth d as a solid line with computational results for contact between (**a**) a sphere with radius $R = 1$ m and a half-space, (**b**) a sphere with radius $R = 2$ m and a half-space, (**c**) two spheres with equal radius $R = 1$ m, (**d**) two spheres with radii $R_1 = 1$ m and $R_2 = 2$ m, (**e**) a cylinder with radius $R = 1$ m and a half-space, (**f**) a cylinder with radius $R = 2$ m and a half-space, (**g**) two crossed cylinders with equal radius $R = 1$ m, and (**h**) two crossed cylinders with equal radius $R = 2$ m.*

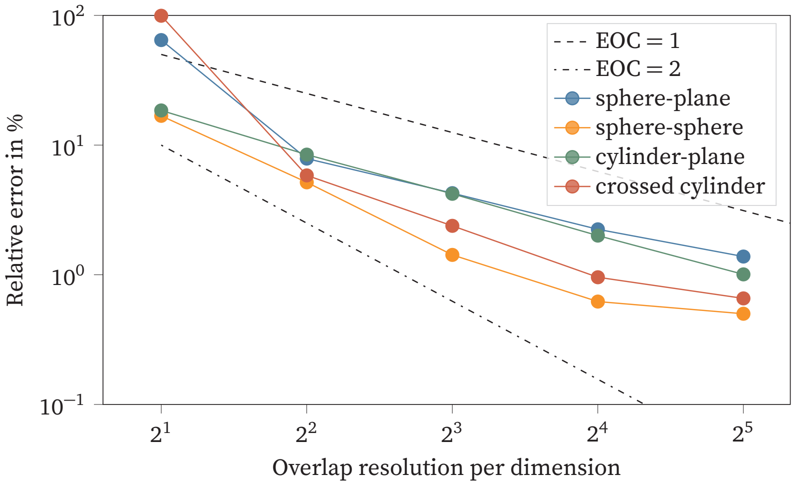


Figure 3.4: *Relative error versus overlap resolution per dimension N_c.*

the average errors of the respective test cases are plotted over the dimension of the overlap N_c and the indentation depth $d = 0$ remains unconsidered. In addition, we compare the data with the experimental order of convergence, EOC = 1 and EOC = 2. As expected, a reduction of the error with increasing resolution is noticeable. Furthermore, we see in large parts of the diagram that a linear slope similar to the EOC = 1 exists, i.e., doubling the resolution halves the error. However, outliers occur because the anisotropic voxel representation in some cases over- and in other cases underestimates curved surfaces.

3.4.2 Cylinder-wall impact

As a further test, the impact of a cylinder on a plane wall is investigated. Since results for this are known from the literature [113], this case is frequently considered [36, 44].

Figure 3.5 shows the basic setup. A cylinder with radius R and height H moves at a constant velocity v_z^- toward the wall located on the x-axis since no fluid is present. Furthermore, there is no friction between the cylinder and the wall.

According to Park [113], the post-impact angular velocity reads

$$\omega_y^+ = \frac{m v_z^- (1+e) r \cos(\alpha+\theta)}{I_{yy} + m r^2 \cos^2(\alpha+\theta)}, \tag{3.33}$$

with the mass of the cylinder m, the signed coefficient of restitution $e = v_z^+/v_z^-$

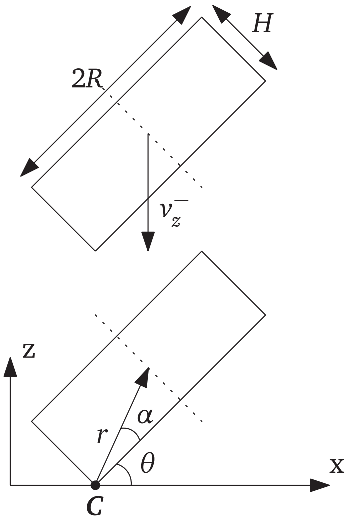


Figure 3.5: *Sketch of the normal cylinder-wall impact test case.*

that depends on the post-impact velocity v_z^+, the moment of inertia with respect to the y-axis I_{yy}, and the distance between the impact point $\boldsymbol{C}$ and the center of the cylinder $r = \sqrt{R^2 + \frac{1}{4}H^2}$. Additionally, two angles influence the outcome. One is the impact angle θ, which can take values between 0°and 90°and the other one is α, which is the angle between the line from the point of impact to the center of the cylinder and the face of the cylinder. There is also a solution for the rebound velocity [113]

$$v_z^+ = \omega_y^+ r \cos(\alpha + \theta) - e v_z^-. \tag{3.34}$$

In the following, we consider a cylinder with a height $H = 0.0053$ m, radius $R = 0.004$ m, mass $m = 3.1 \cdot 10^{-4}$ kg and a moment of inertia $I_{yy} = I_{xx} = 1.96566 \cdot 10^{-9}$ kg m^2 and $I_{zz} = 2.48 \cdot 10^{-9}$ kg m^2. Furthermore, the cylinder settles with an initial velocity of $v_z^- = -1$ m/s. The cylinder's radius is resolved by a resolution of $N = 4$, for the on-lattice contact detection. However, for the rest of the contact treatment, the resolution N_c varies as is visible in Figure 3.6(a) to (b). Additionally, we use time steps of $\Delta t \approx 1.7 \cdot 10^{-6}$ s. When a collision is detected, we reduce the time step size and use $\Delta t_c = \Delta t/10$ instead. Furthermore, we set $c = 0.264$ s/m since this corresponds to a coefficient of restitution of about 0.85, thus $e = -0.85$ applies. The modulus of elasticity is $E = 5 \cdot 10^8$ Pa for both the wall and particle. Following Kodam et al. [36], we set the Poisson's ratio ν both times to 0.35.

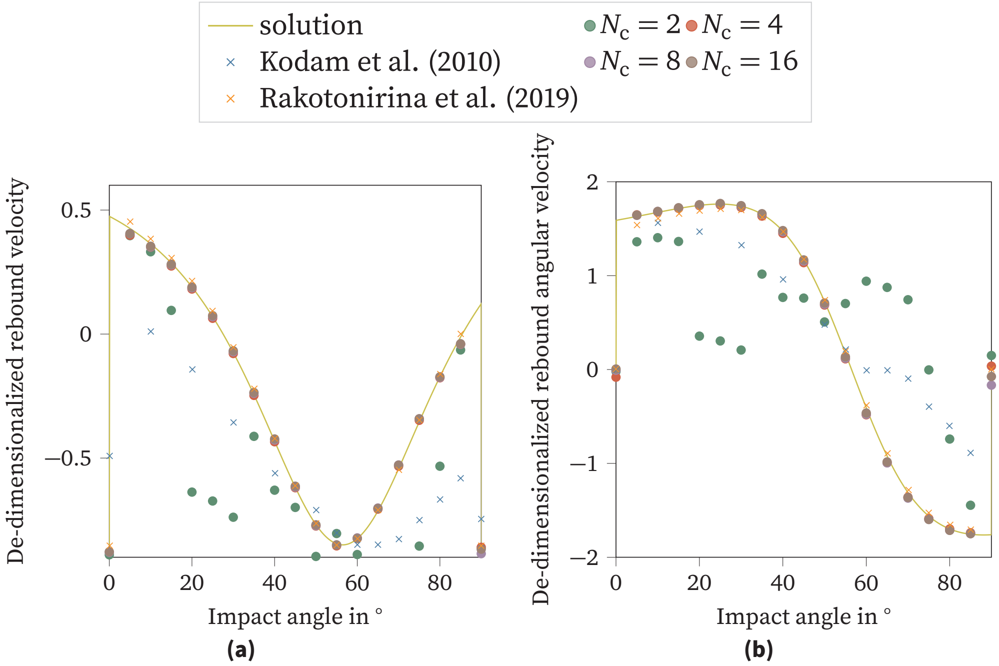


Figure 3.6: *Comparison of the analytical solution and computational results of dimensionless post-impact velocities for the collision of a cylinder and a wall. The plots show (**a**) the rebound velocity v_z^+/v_z^- and (**b**) the rebound angular velocity $r\omega_y^+/v_z^-$ over the impact angle θ.*

In Figure 3.6(a), we compare the resulting de-dimensionalized rebound velocity from our simulations with the analytical solution mentioned before Eq. (3.34) and simulations by Kodam et al. [36] using three layers of glued spheres and Rakotonirina et al. [44] using three glued cylinders. Furthermore, we plot the de-dimensionalized angular velocities versus the analytical solution Eq. (3.33) and numerical results in Figure 3.6(b). In both figures, we consider post-collision values.

We see in both cases that a low resolution, $N_c = 2$, again yield less satisfactory results. However, a good agreement with the comparison data is already obvious from a resolution of $N_c = 4$, which becomes even better with further increasing resolution. It is visible that the simulation data conform to the solid line of the analytical solution. This agreement is better than for the three layers of glued spheres by Kodam et al. [36] illustrated as blue crosses and similar to consideration of three glued cylinders by Rakotonirina et al. [44] illustrated as orange crosses. For the latter, good results are expected, since a cylinder can be perfectly composed of smaller cylinders. For more complex geometries, however, a perfect match may not be possible and the results may decrease in accuracy similar to the glued spheres. In general, this test case shows a very good accuracy of the proposed novel method.

3.5 Particle rebound in viscous fluid

In the following chapter, we deal with a collision between a spherical particle and a resting wall in a viscous fluid to further validate the presented method and to show its potential.

3.5.1 Description

We numerically study the experimental setup by Li [114] in which spherical particles with a radius of 4.75 mm, a density of 7780 kg/m^3 and a moment of inertia of $3.15207 \cdot 10^{-8}$ kg m^2 are released from different heights. We consider the initial heights h_0 of 5.5 mm, 19.6 mm and 35.7 mm to cover the range of experiments. The surrounding fluid is an aqueous glycerol solution that has a density and viscosity of 1230 kg/m^3 and $50.2 \cdot 10^3$ Pa · s, respectively.

In the numerical setup, we consider a closed container with a height of 0.05 m and an equally sized length as well as width of 0.042 m. For the walls, we use a no-slip halfway bounce-back condition [91]. Furthermore, we set the elasticity modulus of the particle to 20 GPa and of the wall to 3 GPa and use the damping constant $c = 0.12$. Additionally, the Poisson's ratios of the particle and wall are 0.33 and 0.24.

We discretize the spatial domain by $\Delta x = 0.3$ mm, and $\Delta t = 5$ µs is the size of the time steps to solve the fluid with HLBM. To solve the equations of motion and calculate the contact force, we use smaller time steps of $\Delta t/1000$. For this, it is necessary to sufficiently resolve the time the particle and the wall are in contact. Also, we use an overlap resolution $N_c = 16$. To avoid tiny channels between the particle and the wall and thus insufficiently studied nanoeffects, we enlarge the particle by $\Delta x/7$ solely for the collision consideration.

In this case, the tangential force is negligible. It therefore requires another benchmark with surrounding fluid for validation of the existing model in future considerations.

3.5.2 Results

A comparison of the results obtained by using the proposed novel method with the experimental results by Li [114] and numerical results by Qiu and Wu [46] is given in Figure 3.7. In this figure, we see the minimum distance h from the falling sphere's surface to the bottom wall plotted over time. We consider three cases, as mentioned before. Different colors highlight these. Thereby, the simulation data are shown as dashed lines and the measurement data by Li [114] as points and the results of Qiu and Wu [46] as crosses.

In all cases, the ball first accelerates from its rest position h_0 towards the bottom wall. Due to varying drop heights, acceleration phases of different lengths are also noticeable. The subsequent contact with the wall causes the spherical particle to move upwards. Therefore, the distance h increases again. However, the height reached after the rebound is smaller than the initial height h_0 before and particles with greater drop height also rebound more strongly and reach a greater heights after the impact. This process is repeated until the ball comes to rest on the ground, as each time, it loses energy.

Overall, a good agreement of our results with the data by Li [114] and Qiu and Wu [46] is visible. Small deviations are particularly noticeable in the first impact from the highest falling height h_0, since the influence of the fluid is greatest there. We can see, for example, that in the same case, at the second impact, the maximum rebound height corresponds to the measured values. Furthermore, deviations between the numerical results of the current simulation and the one of Qiu and Wu [46] are evident in the initial acceleration of the sphere.

3.5.3 Discussion

We have shown that the proposed discrete collision model is capable of an accurate contact treatment for arbitrarily shaped particles. The contact model is essential in the considered case because without it there would be no rebound and only a deceleration of the particle. This reveals the great potential of this novel model for further studies of particulate flows of all kinds.

However, we also see that the method still has further potential for improvement. For example, the different heights after the first rebound of the sphere with an initial height of 35.7 mm in Figure 3.7 are probably due to the assumption of a constant damping factor. Contrary to the assumption, the literature suggests that the damping depends on the initial relative velocity of the two objects in contact.

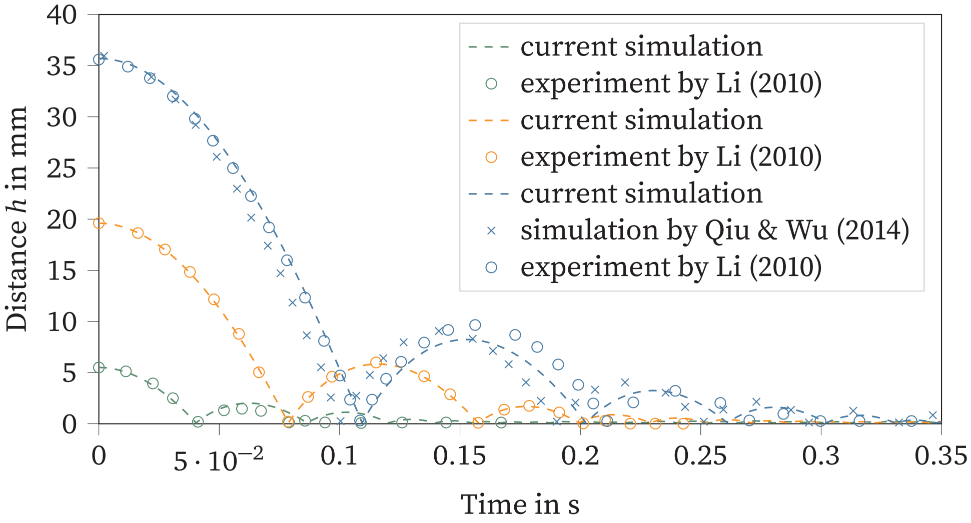


Figure 3.7: *Plot of the shortest distance between a settling sphere's surface and the bottom wall over time.*

The differences between the current and the results by Qiu and Wu [46] in the initial particle acceleration are because of the surface resolved particles in this work and the resulting advantages in the determination of hydrodynamic forces.

It is also evident that it is important to study the mechanics of almost collisions, especially for viscous fluids. This is due to the fact that the formation of nanoscale channels may induce other effects. As we have shown, it is possible to model the collision by a virtual particle enlargement. However, further studies are required to find an optimal parameter which influences the accuracy of the result solely positively.

3.6 Summary and conclusions

In this paper, we propose a novel contact model for complex arbitrary-shaped convex geometries and show that it describes contacts with high accuracy. This is highlighted multiple times in various cases, where we verify the correctness of the contact force and the resulting motion of a particle with and without a fluid present.

In the latter case, we furthermore demonstrate that the proposed method works well with HLBM. However, other PSMs make no difference. With this contact model, it is now possible to simulate complex suspensions precisely and thus to study real-world applications, e.g., hindered settling, extensively.

However, there is also potential for improvement, e.g. we recommend deriving the contact model parameter k directly from the respective shape of the overlap, which further improves the accuracy of the resulting force. Furthermore, future studies of the relationship between the damping factor c and the initial relative velocity have the potential to lead to improvements in the proposed method. In addition, it is significant to model the respective fluid correctly, even with nano-sized channels shortly before the collision. Alternatively, this is solvable by a virtual enlargement of the colliding objects. However, this solution also requires further investigations. Additional studies are also desired for validation of the tangential force in the presence of a (viscous) fluid.

Overall, the novel method suits the previously mentioned challenges well and can be used flexibly with different methods for particulate flows, such as PSMs.

4 Software Concept and Realization

The following chapter summarizes the Lagrangian framework as implemented in the open source software OpenLB [75, 94] for the consideration of discrete particles. A large number of different methods exist for such systems. These methods share the fundamental challenge of efficiently solving the equations of motion while accounting for different physical scenarios and system complexities, such as interactions with the fluid or other particles. To this end, the methods use common data sets that include parameters such as the position of the center of mass, mass, and velocity. However, they differ in the implementation of coupling mechanisms and computational strategies, such as parallelization.

The above results in a software design challenge for a simulation software that includes many different methods, such as OpenLB, as this easily leads to redundancy, making the software difficult to maintain, test, and, in the long run, use. The following chapter therefore introduces a unified modular particle framework that is designed to address the aforementioned challenges by consolidating common data structures and providing a flexible architecture that is easily extensible without the need to modify existing implementations.

In the context of *surface resolved* particles, a key component is the geometry representation introduced in Section 4.1, which is used throughout the code base to represent all geometries, including walls, particles, and other elements. This ensures consistency and efficiency throughout the software. The particle framework that uses this unified geometry representation is explained in Section 4.2, which focuses on *surface resolved* particles, but the current implementation also allows the use of *subgrid scale* particles.

Another major challenge is to ensure software quality in a fast-moving field like CFD. To this end, several measurements are introduced as described in Section 4.3.

While the goal is to cover key aspects and functionalities, a detailed exploration of the software's intricacies is beyond the scope of this thesis. For a complete understanding and detailed instructions on how to use the software, interested readers are referred to the software's user guide [115].

4.1 Geometry Representation

In order to simulate a large number of *surface resolved* particles, an efficient representation of the geometry is required. To achieve this, a hierarchical geometry representation structure has been implemented using object-oriented programming principles. This approach provides a flexible and extensible framework for handling different particle shapes and configurations.

The geometry representation is implemented using a class hierarchy, with a base class serving as the interface for the different shapes, as simplified illustrated in Figure 4.1. This allows easy interchangeability between different geometries. For example, a simple sphere can be easily replaced by a more complex shape, such as a superellipsoid, without changing the overall structure of the simulation. Each derived class in the hierarchy encapsulates the necessary computations specific to its geometry. A prime example of this is the calculation of the signed distance to the particle surface, which is critical to many aspects of the simulation.

The framework supports several analytical geometries, including spheres (`IndicatorSphere3D`), cylinders (`IndicatorCylinder3D`), ellipsoids (`IndicatorEllipsoid3D`), and superellipsoids (`IndicatorSuperEllipsoid3D`). However, its flexibility extends beyond these predefined shapes. It allows the integration of arbitrary geometries through two main approaches. First, STL (stereolithography) files can be used, using the `STLreader`, allowing the incorporation of complex, real-world geometries that can be obtained from computed tomography (CT) scans or created using external computer-aided design (CAD) tools. Second, by using signed distance functions (SDF), it is possible to efficiently modify existing geometries to create more complex shapes. For example, using Boolean operations such as union, intersection, and difference.

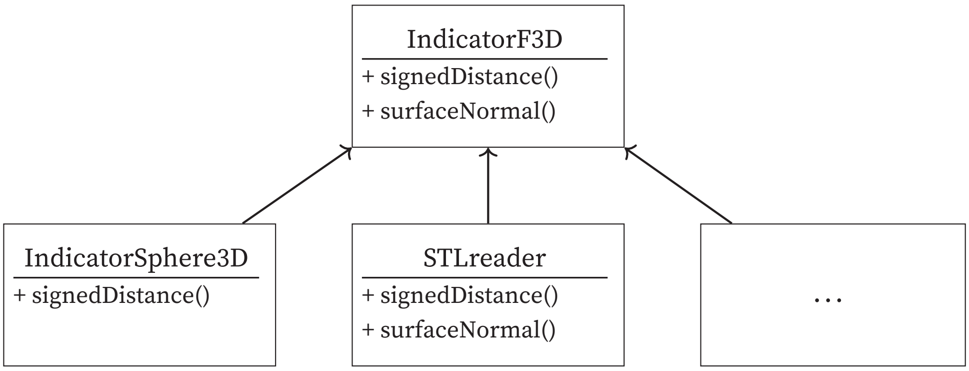


Figure 4.1: *Simplified class hierarchy for 3D geometries.* `IndicatorF3D` *is the base class, with examples of derived classes.*

The spatial derivative of the SDF yields the surface normal, providing an efficient and general method for normal vector computation. This approach is particularly effective for analytic geometries and combined shapes and is therefore used for most geometries. For example, in the simplified example shown in Figure 4.1, the base class `IndicatorF3D` provides the method and the derived class `IndicatorSphere3D` inherits it, thus avoiding redundancy.

For STL-based geometries, an octree structure is used to efficiently evaluate the closest distance to the triangles near the point of interest. Similarly, the surface normal is obtained using the aforementioned octree approach that identifies the normal of the closest triangle. Since the `STLreader` is unique in this respect, it provides its own methods to compute the signed distance and the surface normal, see Figure 4.1.

This comprehensive approach to geometry representation provides the flexibility and efficiency required for advanced particulate flow simulations, allowing the study of a wide range of particle shapes and configurations within a unified framework. Importantly, this representation is also applied in a similar manner to represent other simulation elements such as walls, thereby promoting code reuse and reducing duplication across the code base.

4.2 Discrete Particle Framework

The basic idea of the framework is to separate data and operations, as visualized in Figure 4.2. This means that the particle object solely holds the data, while various operations are performed on this data independently. For instance, the

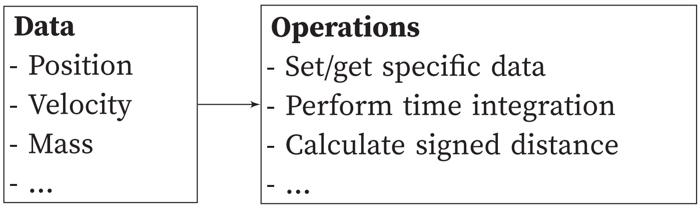


Figure 4.2: *Illustration of the simplified separation of data and operations in the discrete particle framework.*

particle object serves as a container for attributes such as position, velocity, and other properties relevant to the simulation. However, instead of embedding methods within the particle object to manipulate this data directly, a set of separate operations is provided. These operations act on the particle data without being tightly coupled to it, aligning well with the single responsibility and open/closed principles.

Furthermore, the particle data is modular, i.e. it is easy to add or remove data depending on the needs in the simulation by changing the `PARTICLETYPE`. For example, it is easy to replace the particle mass with the density, if necessary. Figure 4.3 visualizes this concept, showing general data such as position and particle validity, but also more specific parameters such as the forcing group and the mobility group. The former consists of the forces acting on the particle, while the latter includes all parameters needed to solve the chosen time integration scheme. The particle is additionally extendable by physical properties, such as mass and density, as well as mechanical properties of the particle, among others.

For easier processing of particles located on the same sub domain inside the computational domain (block), they are stored in the same blockwise container called `ParticleSystem`. In parallel simulations multiple blocks exists, hence they are combined in the `SuperParticleSystem`.

The `ParticleDynamics` define the time integration scheme used to solve the equations of motion, for instance to compute the new position and velocity of the particle. So called particle tasks are constructs that define operations on the particle data. They solve the equations of motion, using the above defined `ParticleDynamics`, but also perform the coupling and communication. These tasks alone would be sufficient, but to avoid repeating iterations over all particles, the `ParticleManager` is used. It allows a list of tasks to be performed in a predefined order per particle, improving computational performance, especially for large particle systems with tens of thousands of particles.

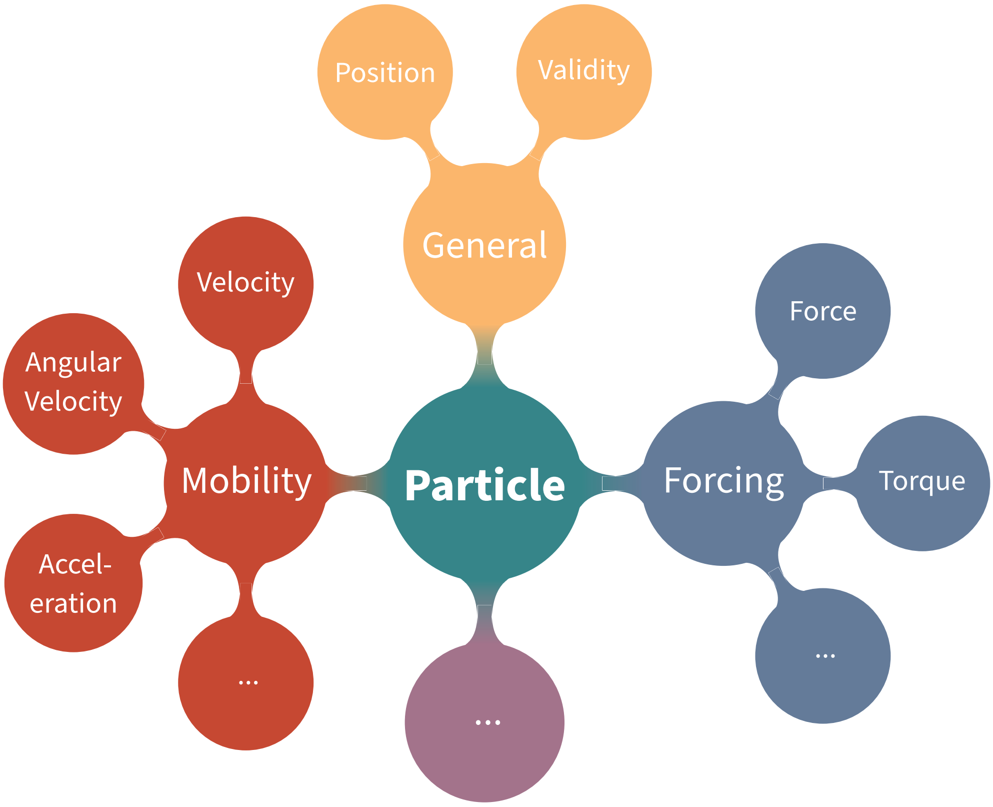


Figure 4.3: *Visualization of the concept of modular particle data that allows the addition and removal of data depending on the needs of the used model.*

Because introducing particles into such a system is potentially complex, factory methods are provided for easy particle creation, leading to greater ease of use and therefore sustainability.

After the abstract explanation of the various parts above, details of their explicit application follow.

4.2.1 Particle Data

The data a particle stores and provides for particle tasks are defined by the `PARTICLETYPE`. For easier usage of frequently used configurations, several aliases are predefined in `src/particles/descriptors/particleDescriptorAlias.h`. Those aliases easy to use and replace as shown in Listing 4.1.

The main difference between the types shown in Listing 4.1 is in the fields

```
      // PARTICLETYPE for surface resolved particles
      typedef ResolvedParticle3D PARTICLETYPE;

      // PARTICLETYPE for \acl{sgs} particles
      typedef SubgridParticle3D PARTICLETYPE;
```

Listing 4.1: *Usage of particle types examplified on surface resolved and subgrid scale particles*

they provide. The *surface resolved* particle requires more information for a proper description, including the surface data, while the *subgrid scale* particle is assumed to be a rotation invariant sphere and can therefore neglect rotation.

4.2.2 Class ParticleSystem

The `ParticleSystem` is a basic object because it stores all data about the particles in blockwise containers. Therefore, it is only used explicitly when running sequential simulations. In simulation setups that divide the domain into several subregions, it is part of the `SuperParticleSystem` (cf. Section 4.2.3).

The use of the `ParticleSystem` is exemplified during the initialization in Listing 4.2. First, the initially empty `ParticleSystem` is created with the desired `PARTICLETYPE` (cf. Section 4.2.1). Second, the `ParticleSystem` is added to the `ParticleManager` (cf. Section 4.2.5) to make the data available during the task execution. Third, the particle dynamics are defined. In this example, the `VerletParticleDynamics` (cf. Section 4.2.4) is used. Finally, two particles are created and added to the `ParticleSystem` using factory methods.

4.2.3 Class SuperParticleSystem

To use OpenLB's domain decomposition approach for *subgrid scale* and *surface resolved* particles, the only necessary change is to use the `SuperParticleSystem`, as Listing 4.3 shows. It is created by passing the `SuperGeometry`, which contains all the information about the lattice decomposition. Using this information, the `SuperParticleSystem` creates and holds a `ParticleSystem` (cf. Section 4.2.2) for each subdomain.

4.2.4 Particle Dynamics

The particle dynamics define the time integration scheme used to solve the equations of motion using the particle data (cf. Section 4.2.1) and various forces

```
      // Create ParticleSystem
      ParticleSystem<T,PARTICLETYPE> particleSystem;

      // Create ParticleManager
      ParticleManager<T,DESCRIPTOR,PARTICLETYPE> particleManager(
        particleSystem, superGeometry, sLattice, converter );

      // Create and assign particle dynamics
      particleSystem.defineDynamics<
        VerletParticleDynamics<T,PARTICLETYPE>>();

      // Define size of smooth boundary (scalar)
      T epsilon = 0.5 * converter.getConversionFactorLength();

      // Create Particle 1 - Sphere
      // sphereCenter  is a three-dimensional vector
      // sphereRadius  is a scalar
      // sphereDensity is a scalar
      creators::addResolvedSphere3D( particleSystem, sphereCenter,
        sphereRadius, epsilon, sphereDensity );

      // Create Particle 2 - Cuboid
      // cubeCenter  is a three-dimensional vector
      // cubeExtend  is a three-dimensional vector
      // cubeDensity is a scalar
      creators::addResolvedCuboid3D( particleSystem, cubeCenter,
        cubeExtend, epsilon, cubeDensity );
```

Listing 4.2: *Initialization of particle-related data and objects in sequential simulations.*

```
      // Create SuperParticleSystem
      SuperParticleSystem<T,PARTICLETYPE> particleSystem(
          superGeometry);

      // Create ParticleManager
      ParticleManager<T,DESCRIPTOR,PARTICLETYPE> particleManager(
        particleSystem, superGeometry, sLattice, converter );

      // Create and assign particle dynamics
      particleSystem.defineDynamics<
        VerletParticleDynamics<T,PARTICLETYPE>>();

      // Define size of smooth boundary (scalar)
      T epsilon = 0.5 * converter.getConversionFactorLength();

      // Create Particle 1 - Sphere
      // sphereCenter  is a three-dimensional vector
      // sphereRadius  is a scalar
      // sphereDensity is a scalar
      creators::addResolvedSphere3D( particleSystem, sphereCenter,
      sphereRadius, epsilon, sphereDensity );

      // Create Particle 2 - Cuboid
      // cubeCenter  is a three-dimensional vector
      // cubeExtend  is a three-dimensional vector
      // cubeDensity is a scalar
      creators::addResolvedCuboid3D( particleSystem, cubeCenter,
      cubeExtend, epsilon, cubeDensity );
```

Listing 4.3: *Initialization of particle-related data and objects when using a decomposed domain.*

(cf. Sections 4.2.6 and 4.2.7) as input, resulting in an updated velocity, angular velocity, position, and orientation.

In OpenLB, `VerletParticleDynamics`, which refer to the velocity Verlet algorithm [108, 109], are commonly used. As the velocity Verlet algorithm stands out for its superior second-order accuracy [116], stability [117], and energy conservation properties [118] compared to simpler numerical procedures such as the forward Euler method. Its symplectic nature ensures reliable long-term behavior [119], making it a preferred choice for capturing the intricate dynamics of complex systems.

As shown in Listings 4.2 and 4.3, particle dynamics are defined on the `ParticleSystem` (cf. Section 4.2.2) or `SuperParticleSystem` (cf. Section 4.2.3).

4.2.5 Class ParticleManager

Though optional, the `ParticleManager` provides a structured and readable way to use particle-related tasks in a predefined order with its `execute()` method, as shown in Listings 4.4 and 4.5. The `ParticleManager` also takes care of combining tasks into a single particle loop, improving performance. To this end, the `ParticleManager` is first created with access to particle, lattice and setup specific data, see Listings 4.2 and 4.3.

4.2.6 Two-Way Coupling (Surface Resolved Particles)

To realize simulations considering two-way coupled surface resolved particles without using the particle decomposition scheme, the tasks outlined in Listing 4.4 must be called, for example using the `ParticleManager` (cf. Section 4.2.5).

In this example, `couple_lattice_to_particles` refers to the operation that computes the hydrodynamic forces acting on the particles, while `process_dynamics` solves the equations of motion using the defined particle dynamics (cf. Section 4.2.4). The last task `couple_particles_to_lattice` maps the particle to the fluid mesh for back coupling. Note that to actually process the back coupling, a specific lattice dynamics must be chosen that allows for such interactions, e.g. `PorousParticleBGKdynamics`.

The particle decomposition scheme [72] improves the computational speed, but slightly increases the complexity because it adds the need for communication steps that must be called explicitly, see Listing 4.5. Since the particle's surface may be on multiple subdomains, the partial surface forces must be communicated to the process responsible for solving the equations of motion

```
particleManager.execute<
  couple_lattice_to_particles<T,DESCRIPTOR,PARTICLETYPE>,
  process_dynamics<T,PARTICLETYPE>,
  couple_particles_to_lattice<T,DESCRIPTOR,PARTICLETYPE>
>();
```

Listing 4.4: *Execution of particle tasks to consider two-way coupled surface resolved particles without particle decomposition.*

```
particleManager.execute<
  couple_lattice_to_particles<T,DESCRIPTOR,PARTICLETYPE>,
  communicate_surface_force<T,PARTICLETYPE>,
  process_dynamics<T,PARTICLETYPE>,
  update_particle_core_distribution<T, PARTICLETYPE>,
  couple_particles_to_lattice<T,DESCRIPTOR,PARTICLETYPE>
>();
```

Listing 4.5: *Execution of particle tasks to consider two-way coupled surface resolved particles with particle decomposition.*

using `communicate_surface_force`. After solving the equations of motion, the particles have updated positions and orientations, which potentially resulting in different subdomain intersections and thus assignments to process units. However, only one process unit has access to the latest particle data, so it communicates the particle distribution using `update_particle_core_distribution`.

4.2.7 Discrete Contact Model (Surface Resolved Particles)

The consideration of solid-solid interactions results in the need to store more data and provide more operations. First, to remove unnecessary dependencies and to allow easy replacement of the contact model, contact types for particle-wall and particle-particle contacts are introduced. Listing 4.6 shows the contact types and their definition for the discrete contact model [71], which computes contact forces for arbitrary shapes from an overlapping volume. These contact types define the dynamic data that each individual contact stores, including the IDs of the objects in contact, the minimum and maximum coordinates of the overlap region, a damping factor, and the ID of the process unit responsible for calculating and applying the resulting contact forces. Note that even though the contact model works for arbitrary shapes, the currently implemented contact detection is limited to convex shapes.

```
   typedef ParticleContactArbitraryFromOverlapVolume<T,
     PARTICLETYPE::d> PARTICLECONTACTTYPE;
   typedef WallContactArbitraryFromOverlapVolume<T,
     PARTICLETYPE::d> WALLCONTACTTYPE;
```

Listing 4.6: *Definition of types for particle-particle and particle-wall contacts.*

```
  ContactContainer<T, PARTICLECONTACTTYPE, WALLCONTACTTYPE>
     contactContainer;
```

Listing 4.7: *Creation of a contact container that stores individual contacts after rough detection for later processing.*

The `ContactContainer` provides a single container to store all particle-particle and particle-wall contacts therefore needing the aforementioned contact type on creation, see Listing 4.7.

When calculating contact forces, it is important to consider parameters for the material combination at hand, such as the effective Young's modulus and coefficients of friction. Since these data are constant for the material pairs, they are stored in the lookup table `ContactProperties`. This reduces the memory load because the number of contacts is potentially very high, especially in tight packings, i.e. high particle numbers and volume fractions. Materials in this context are user-defined IDs that allow common mechanical properties to be used during contact treatment. For example, the particles in the considered system are made of the same rubber, so they have the same material `particleContactMaterial`, while the walls are made of steel, a different material, so they are identified by another material ID `wallContactMaterial`. The Listing 4.8 illustrates the use of the `ContactProperties`. It is important to note that for each material pair, `ContactProperties.set(...)` must be called with the proper parameters to populate the lookup table with the correct data.

Table 4.1 gives concrete details about parameters that are stored in individual contacts and parameters that are shared in the above lookup table. The examples show that parameters that differ for individual contacts are associated with the corresponding contact object, while data that is constant for material combinations is located in the shared lookup table.

While the above mentioned particle systems (cf. Sections 4.2.2 and 4.2.3) contain the particles, `SolidBoundary` refers to single solid boundary objects like walls, which are stored in a simple vector of `solidBoundaries`. This is nec-

```
ContactProperties<T, NUMBER_OF_MATERIALS> contactProperties;
T effectiveYoungsModulus = evalEffectiveYoungModulus(
  youngsModulusParticle,
  youngsModulusWall,
  poissonRatioParticle,
  poissonRatioWall ),
contactProperties.set(
  particleContactMaterial,
  wallContactMaterial,
  effectiveYoungsModulus,
  coefficientOfRestitution,
  coefficientKineticFriction,
  coefficientStaticFriction );
```

Listing 4.8: *Initialization of contact properties for a pair of materials.*

Table 4.1: *Comparison of examples of contact-specific (`Contact`) and material-dependent (`ContactProperties`) data.*

`Contact`	**`ContactProperties`**
Bounding box of the contact region	Effective Young's modulus
Damping factor	Coefficient of restitution
IDs of the objects in contact	Coefficient of static friction
Responsible processing unit	Coefficient of kinetic friction

essary to make the solid boundary information, such as geometry, available and easily accessible during contact handling, since the `SolidBoundary` stores the aforementioned material ID used for identification and lookup. Listing 4.9 gives an example of creating a single `SolidBoundary` and adding it to a container named `solidBoundaries`. Note that in addition to the contact material, called `wallContactMaterial` in this example, a lattice material `wallLatticeMaterialNumber` is also passed to and stored in the object. The lattice material allows the assignment of regions on the lattice to the wall for an initial rough particle-wall contact detection. Since the materials are used for identification in the context of walls, each wall in the container must have different materials. To handle complex geometries of the same material, the OpenLB functor concept can be used to combine different indicators, or different IDs can be used but the `ContactProperties` can be populated with the same properties.

The aforementioned rough contact detection is performed during the step that maps the particles onto the lattice in order to speedup computations. Hence

```
std::unique_ptr<IndicatorF3D<T>> wallIndicator =
  std::make_unique<IndicatorCuboid3D<T>>( wallExtend,
                                          wallOrigin );
solidBoundaries.push_back(SolidBoundary<T, PARTICLETYPE::d>(
  std::move(wallInd), wallLatticeMaterialNumber,
  wallContactMaterial) );
```

Listing 4.9: *Creation of a solid boundary from an indicator and addition to a vector of multiple solid boundaries.*

```
coupleResolvedParticlesToLattice<T, DESCRIPTOR,
  PARTICLETYPE, PARTICLECONTACTTYPE, WALLCONTACTTYPE>(
    particleSystem, contactContainer, superGeometry,
    superLattice, converter, solidBoundaries );
```

Listing 4.10: *Mapping of particles to the fluid lattice with simultaneous rought contact detection.*

the task `couple_particles_to_lattice` is replaced by the method call shown in Listing 4.10 in order to map the particles to the fluid lattice, while simultaneously performing a rough contact detection and updating the `ContactContainer`.

Before solving the equations of motion using the task `process_dynamics`, the contact forces must be computed and pplied to the particles, which is done by the function call shown in Listing 4.11.

4.3 Software Quality Assurance

This section encapsulates a variety of practices dedicated to upholding the reliability, efficiency, and efficacy of the developed software. While it generally encompasses broader concepts such as quality control, enhancement, documentation [115], and tools, it specifically delves into the continuous integration

```
processContacts<T, PARTICLETYPE, PARTICLECONTACTTYPE,
  WALLCONTACTTYPE, ContactProperties<T, NUMBER_OF_MATERIALS>>(
    particleSystem, solidBoundaries,
    contactContainer, contactProperties,
    superGeometry );
```

Listing 4.11: *Computing contact forces for all found contacts.*

(CI) process, unit tests (cf. Section 4.3.1), and benchmark tests (cf. Section 4.3.2).

In order to enable a continuous integration (CI) of incremental improvements, several tasks are performed automatically before merge request are introduced into the main branch. These include basic steps like sanitizing, building the source code with different compilers, and extracting the documentation. However, also unit tests (cf. Section 4.3.1) and validation using established benchmarks (cf. Section 4.3.2) are processed to ensure quantitatively correct results across the covered code base.

Automating these checks is important because it frees up human resources for more important tasks, such as adding improvements. It also builds confidence in the software among developers and users alike.

4.3.1 Unit Tests

Unit testing is the cornerstone of software validation, focusing on individual code components to ensure their functionality in isolation. By subjecting these units to rigorous testing, it is easier to find and fix errors early, which promotes code reliability and maintainability. In this section, selected unit tests that provide insight into the effectiveness and performance of key software components are explored.

The core of the particle framework relies heavily on SDF, which are crucial for accurate geometry representation (cf. Section 4.1). To ensure its accuracy and to detect problems immediately, each SDF is rigorously validated against predefined configurations. This validation involves evaluating the functions responsible for generating SDF at various points for all implemented geometries, including spheres, cubes, cylinders, etc., and comparing their outputs with expected results derived from analytical functions.

Similarly, other essential functionalities such as functions to set and get data from a particle (cf. Section 4.2.1) are constantly tested. Here, it is expected that the data that was set is also obtained from the same particle.

Following a logic similar to the one described above, many parts of the code base are tested continuously, including the coupling tasks. While the testing of interactions may not strictly fit the definition of unit tests due to their reliance on multiple functionalities such as SDF, they remain essential for comprehensive validation. For example, particle-particle and particle-wall interactions are evaluated against known analytical solutions [111, 112], ensuring the accuracy of the underlying logic despite their interconnected nature.

Note that all tests run solely sequential, hence testing communication steps cannot be performed during this stage. However, for the particle simulations, proper communication is ensured during the benchmark tests, as outlined in Section 4.3.2.

4.3.2 Benchmark Tests

Unlike unit tests, the benchmarks considered in this section are much more complicated, and correct results depend heavily on the interplay of multiple functionalities. In addition, they offer and have been used in test-driven development to improve computational performance. They help identify performance bottlenecks and efficiently test and evaluate alternatives.

In the context of particulate flow simulations, two benchmarks have been introduced: the behavior of a sphere in a Couette flow (cf. Section 4.3.2.1) and a cylinder wall impact test(cf. 4.3.2.2). Both cases are tested both sequentially and in parallel.

4.3.2.1 Particle in Couette Flow

To ensure the correctness of fluid-particle interactions, a simulation of a particle submerged in a Couette flow is considered. To this end, a periodic channel spanning 0.3 m in the x-direction and 0.25 m in the z-direction, featuring periodic boundaries in both dimensions, as shown in Figure 4.4 is used. In the y-direction the two opposing walls are 0.1 m apart. The upper wall has a velocity of $u_{\mathrm{f,max}}$ and the lower wall of $u_{\mathrm{f,min}} = -u_{\mathrm{f,max}}$. In the recurring tests, two velocities, 0.005 m/s and 0.01 m/s, are considered.

At the center of the channel, a spherical particle with a radius of 15cm is positioned. The sphere diameter is resolved with 10 cells and the lattice relaxation time is $\tau = 1$. Both the fluid and the particle share the density of 1200 kg/m^3. The fluid's viscosity is 1/120 m^2/s.

The simulations consider a time frame of 0.4 s. The x-position of the particle is varied. It is positioned in the center of the x-direction and at the periodic boundary, to also ensure the correctness of the periodic boundary. The different positions and velocities make a total of four setups continuously evaluated for correctness.

In the described setup, the expected angular velocity reads [120]

$$\omega_{\mathrm{p}} = \frac{u_{\mathrm{f,max}}}{H}. \tag{4.1}$$

The tests are considered successful if the relative error is 3% or less.

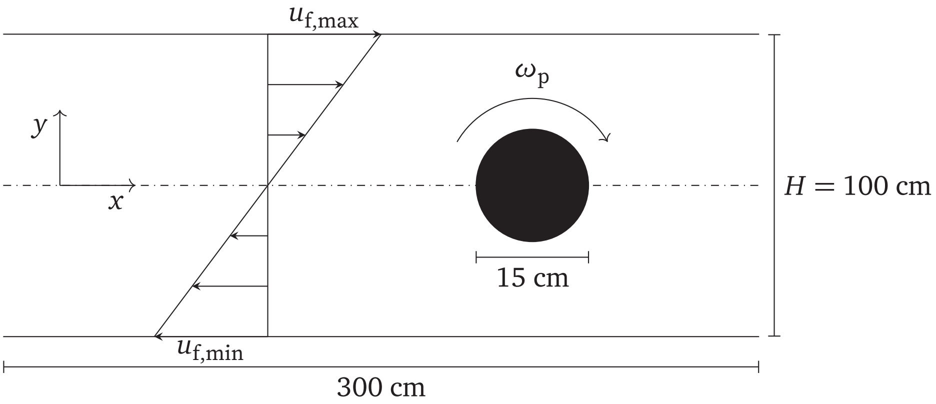


Figure 4.4: *Sketch of the test case considering a single sphere inside a Couette flow as a projection onto the x-y plane.*

The above setup was chosen over a simple sedimentation because the required simulation time is shorter. In addition, the rotation is checked in this context, i.e. force and torque are tested, while a simple sedimentation is less dependent on the correct rotational motion.

4.3.2.2 Cylinder-Wall Impact

The goal of this automated benchmark test is to ensure the correctness of contact detection and solid-solid interaction calculations.

This setup follows the one introduced in Chapter 3. However, some parameters have been changed in order to reduce the computational effort. The cylinder settles with an initial velocity of $v_{\mathrm{z}}^{-} = -0.1$ m/s. The modulus of elasticity is $E = 5 \cdot 10^{8}$ Pa and the Poisson's ratio is $\nu = 0.35$ for both the wall and the cylinder. In addition, time steps of $\Delta t = 1.9 \cdot 10^{-5}$s are used, while the contact treatment time steps are refined to $\Delta t_{\mathrm{c}} = \Delta t/10$.

5
Particle Shape Effects in Hindered Settling

This chapter has been published as

Marquardt, J. E. et al. "A novel model for direct numerical simulation of suspension dynamics with arbitrarily shaped convex particles." In: Computer Physics Communications *304 (2024), p. 109321.* ISSN*: 0010-4655.* DOI*: 10.1016/j.cpc.2024.109321*

5.1 Introduction

Understanding the dynamics of suspensions, which consist of solid particles dispersed in a liquid [11], plays a pivotal role in revealing significant phenomena that arise from the intricate interplay between fluid-particle and particle-particle interactions. These studies are particularly important in the context of industrial processes, as they have the potential to advance equipment design, improve operational practices, and lead to improvements in efficiency, throughput, and product quality. Examples include improving separation efficiencies and reducing damage to the particular phase, which is highly important in food processing.

The study of suspension dynamics is of tremendous importance, but associated with inherent challenges. This is exemplified by hindered settling in a viscous fluid, a seemingly straightforward process involving the settling of multiple particles under the influence of gravity. However, the interaction between these particles results in terminal velocities that differ from those observed in single particle sedimentation [56, 76–81, 121] and clustering [51, 54, 55], indicating high complexity.

All of the above studies focused on spherical particles, whereas the characterization of non-spherical or irregular particles has received less attention. Yet, when examined, the influence of particle shape on the average settling velocity has been found to be significant, as was demonstrated by studies on cubes [20], rod-like particles [21], sand grains [22], and fibers [23]. Unfortunately, experimental studies present significant challenges in terms of control, evaluation, and determination of specific parameters, such as particle shape. Consequently, numerical studies are needed to obtain additional data and a deeper understanding of the underlying dynamics. Such studies require the use of sophisticated models capable of accurately accounting for the complex interplay between fluids and particles, as well as the emergence of solid-solid interactions as the particle volume fraction increases [30]. However, when considering complex particle shapes, the above remains a formidable challenge, as discussed below.

Several approaches to study complex particle systems exist. A commonly used option is the discrete element method (DEM). In DEM, arbitrary particle shapes are often achieved by combining spheres [41, 42]. However, the multi-sphere approach can lead to inaccuracies when using a limited number of spheres or can become computationally expensive, as the number of spheres increases. An alternative is to use other convex shapes [43–45] or level-set DEM [122–124]. However, this significantly increases the complexity, while the coupling with the surrounding viscous fluid remains challenging. An approach to fluid-particle coupling uses simple spherical geometries [46, 47], which leads to inaccuracies. Another approach omits the back-coupling to the fluid [48], which, however, is essential especially at high particle volume fractions [30].

The immersed boundary method (IBM) is a widely used and promising approach to studying complex particle systems. In IBM, the particle surface is represented by Lagrangian points [58]. This method offers high flexibility, as it can be coupled with different fluid solvers, including the finite element method and the lattice Boltzmann method (LBM). IBM also exhibits a high degree of accuracy, as the interaction between these Lagrangian points and the fluid grid is independent. Four-way coupled simulations are also feasible, since a contact model suitable for arbitrary shapes exists [59]. However, disadvantages of IBM are the frequent and computationally expensive interpolations.

An alternative LBM-based approach is the partially saturated method (PSM) proposed by Noble and Torczynski [61]. Its derivatives were used extensively in the investigation of particle flows [62, 63]. A notable derivative of PSM is the homogenized lattice Boltzmann method (HLBM) introduced by Krause et al. [66], which has been successfully used to study different particle shapes [67, 68, 86–88]. Particularly noteworthy are studies on the shape-dependent sedimentation behavior of single particles [69]. With the recent development of a compatible contact model [71], similar investigations of shape-dependent hindered settling have become possible. However, the discrete contact model proposed is associated with a major computational burden due to frequent distance calculations, which limits its application to systems with a few particles rather than systems with hundreds or thousands of particles. Despite the introduction of an improved particle decomposition scheme [73], the contact treatment remains the bottleneck in terms of computational efficiency. Therefore, improving its parallel performance is crucial to enabling simulations with large particle numbers at high particle volume fractions.

While parallelization strategies for the simulation of particulate flows have been proposed in literature [125–127], they at best inadequately consider particle surface intersections with subdomains within the domain decomposition, which is important for direct numerical simulations (DNS) as it would otherwise lead to inaccurate capture of fluid-solid and solid-solid interactions. In addition, they typically consider only simple regular shapes, such as spheres. Therefore, a new and efficient parallelization approach for DNS that can handle the increased complexity due to a large number of surface resolved particles with complex shapes is important.

The primary goal of this work is to enable DNS of arbitrarily shaped convex particle collectives. To this end, we propose a novel and improved parallelization strategy for the discrete contact model. This advancement is crucial to overcoming the computational limitations associated with the discrete contact model in the simulation of large-scale systems. Furthermore, this paper seeks to address the challenge of incorporating periodic boundaries into the discrete contact model and it aims to demonstrate the necessity of an explicit contact model when using PSMs, particularly at high particle volume fractions. As a demonstration of the new possibilities, we numerically study the hindered settling of spheres and cubes, leading to new insights into their swarm settling behavior.

The subsequent sections of this paper are structured as follows. Section 5.2 presents the models employed to describe the fluid and particle behavior, while Section 5.3 discusses the numerical techniques utilized to solve the model system. In Section 5.4, we outline the proposed parallelization strategy, which is then applied to investigate hindered settling in Section 5.5. Lastly, Section 5.6

summarizes the key findings and conclusions drawn from this study.

5.2 Modeling

To account for the fluids, particles, and interactions between and within the components, we use the same models as in previous related works [71, 73]. The following sections give a brief overview of these models.

5.2.1 Fluid

Throughout this work, we consider incompressible fluids. Therefore, the Navier–Stokes equations are given by

$$\begin{aligned}\frac{\partial \boldsymbol{u}_\mathrm{f}}{\partial t} + (\boldsymbol{u}_\mathrm{f} \cdot \nabla)\,\boldsymbol{u}_\mathrm{f} - \frac{\eta}{\rho_\mathrm{f}}\Delta \boldsymbol{u}_\mathrm{f} + \frac{1}{\rho_\mathrm{f}}\nabla p &= \frac{\boldsymbol{F}_\mathrm{f}}{\rho_\mathrm{f}},\\ \nabla \cdot \boldsymbol{u}_\mathrm{f} &= 0,\end{aligned} \tag{5.1}$$

where p represents pressure, t denotes time, $\boldsymbol{F}_\mathrm{f}$ signifies the total of all forces acting on the fluid, and $\boldsymbol{u}_\mathrm{f}$, ρ_f, η represent the fluid's velocity, density, and dynamic viscosity, respectively.

5.2.2 Particle

Newton's second law of motion is the basis for our consideration of particles. Hence, translation is governed by

$$m_\mathrm{p}\frac{\partial \boldsymbol{u}_\mathrm{p}}{\partial t} = \boldsymbol{F}_\mathrm{p}, \tag{5.2}$$

while rotation is described by

$$\boldsymbol{I}_\mathrm{p}\frac{\partial \boldsymbol{\omega}_\mathrm{p}}{\partial t} + \boldsymbol{\omega}_\mathrm{p} \times (\boldsymbol{I}_\mathrm{p} \cdot \boldsymbol{\omega}_\mathrm{p}) = \boldsymbol{T}_\mathrm{p}. \tag{5.3}$$

In these equations, m_p, $\boldsymbol{I}_\mathrm{p}$, $\boldsymbol{u}_\mathrm{p}$, and $\boldsymbol{\omega}_\mathrm{p}$ represent the mass, moment of inertia, velocity, and angular velocity of the particle, respectively. The parameters $\boldsymbol{F}_\mathrm{p}$ and $\boldsymbol{T}_\mathrm{p}$ correspond to the total force and torque acting on it. Finally, the subscript p indicates that above variables refer to the particle's center of mass.

5.2.3 Contact

To account for interactions of even complex geometries, we employ a model introduced by Nassauer and Kuna [105]. Here, the normal contact force is given by

$$\boldsymbol{F}_{\mathrm{c,n}} = \frac{4}{3\pi} \boldsymbol{n}_{\mathrm{c}} E^* \sqrt{V_{\mathrm{c}} d} \left(1 + c \dot{d}_{\mathrm{n}}\right), \tag{5.4}$$

where E^* represents the effective Young's modulus, V_{c} denotes the overlap volume, d represents the indentation depth, c is a damping factor, and $\dot{d}_{\mathrm{n}}$ is the magnitude of the relative velocity between two bodies in contact along the contact normal $\boldsymbol{n}_{\mathrm{c}}$.

The effective Young's modulus, E^*, is given by

$$E^* = \left(\frac{1-\nu_{\mathrm{A}}^2}{E_{\mathrm{A}}} + \frac{1-\nu_{\mathrm{B}}^2}{E_{\mathrm{B}}}\right)^{-1}, \tag{5.5}$$

where E_{A} and ν_{A} represent the Young's modulus and Poisson's ratio of an object A, while E_{B} and ν_{B} denote the Young's modulus and Poisson's ratio of another object B. To correlate the damping factor c to the coefficient of restitution e, we follow Carvalho and Martins [106] and use

$$c = \begin{cases} 1.5\frac{(1-e)(11-e)}{(1+9e)u_0} & \text{for } u_0 > 0 \\ 0 & \text{for } u_0 \leq 0 \end{cases}, \tag{5.6}$$

where u_0 is the magnitude of the relative velocity at the initial contact [88].

Tangential forces arise as a result of friction, which is typically influenced by the normal force, along with the coefficients of static and kinetic friction, denoted as μ_{s} and μ_{k}, respectively. Following a similar rationale, Nassauer and Kuna [105] describe the tangential force using the equation

$$\boldsymbol{F}_{\mathrm{c,t}} = -\frac{\boldsymbol{u}_{\mathrm{AB,t}}(\boldsymbol{x}_{\mathrm{c}})}{||\boldsymbol{u}_{\mathrm{AB,t}}(\boldsymbol{x}_{\mathrm{c}})||} \left(\left(2\mu_{\mathrm{s}}^* - \mu_{\mathrm{k}}\right) \frac{a^2}{a^4+1} + \mu_{\mathrm{k}} - \frac{\mu_{\mathrm{k}}}{a^2+1}\right) ||\boldsymbol{F}_{\mathrm{c,n}}||, \tag{5.7}$$

with

$$\mu_{\mathrm{s}}^* = \mu_{\mathrm{s}} \left(1 - 0.09 \left(\frac{\mu_{\mathrm{k}}}{\mu_{\mathrm{s}}}\right)^4\right), \tag{5.8}$$

and

$$a = \frac{||\boldsymbol{u}_{\mathrm{AB,t}}(\boldsymbol{x}_{\mathrm{c}})||}{u_k}. \tag{5.9}$$

$\boldsymbol{u}_{\mathrm{AB,t}}$ represents the relative tangential velocity at the contact point $\boldsymbol{x}_{\mathrm{c}}$ between objects A and B and u_k is a model parameter that defines the velocity at which the transition from static to kinetic friction occurs. In this work, we use $u_k = 0.001\mathrm{m/s}$.

Note that the tangential force model neglects stiction, and thus in cases where it plays a significant role, more sophisticated frictional force models become imperative [128].

5.3 Numerical methods

The following section introduces the numerical methods used to solve the applied models. It is important to emphasize that all quantities discussed in Sections 5.3.1 and 5.3.2 are given in lattice units.

5.3.1 Lattice Boltzmann Method

To solve the incompressible Navier–Stokes equations, the lattice Boltzmann method (LBM) [65, 90, 91] is an established and powerful option. LBM is a mesoscopic approach that discretizes both space and time by representing fluid flow in terms of particle distributions. Therefore, when we refer to particles in the remainder of this section, we are referring to fluid particles. These particle distributions, denoted as $f_i(\boldsymbol{x}, t)$, represent the probability of finding particles with the discrete velocity $\boldsymbol{c}_i$ at a position $\boldsymbol{x}$ and time t. A variety of velocity sets have been proposed and discussed in literature [65, 90]. In this paper, we specifically adopt the D3Q19 velocity set, considering 19 velocities in 3 spatial dimensions.

The LBM algorithm provides a numerical solution for the underlying fluid flow by iteratively updating the particle distributions in two steps. First, the collision step accounts for the local interparticular interactions and reads

$$f_i^*(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) + \Omega_i(\boldsymbol{x}, t) + S_i(\boldsymbol{x}, t), \tag{5.10}$$

where f_i^* is the post-collision distribution, Ω_i is the collision operator, and S_i is an optional source term. Second, the propagation step ensures the distribution of particles to neighboring lattice nodes and is given by

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i, t + 1) = f_i^*(\boldsymbol{x}, t), \tag{5.11}$$

for $\Delta t = \Delta x = 1$, which refer to the time step size and the grid spacing, respectively.

In this work, we use the Bhatnagar–Gross–Krook (BGK) collision operator [92]

$$\Omega_i(\boldsymbol{x},t) = -\frac{1}{\tau}\left(f_i(\boldsymbol{x},t) - f_i^{\mathrm{eq}}(\rho_{\mathrm{f}}, \boldsymbol{u}_{\mathrm{f}})\right), \tag{5.12}$$

with the relaxation time $\tau = 3(\eta/\rho_{\mathrm{f}}) + 0.5$ to relax the particle distributions towards an equilibrium f_i^{eq}. This equilibrium is quantified by the Maxwell–Boltzmann distribution and reads

$$f_i^{\mathrm{eq}}(\rho_{\mathrm{f}}, \boldsymbol{u}_{\mathrm{f}}) = w_i \rho_{\mathrm{f}} \left(1 + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}_{\mathrm{f}}}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}_{\mathrm{f}})^2}{2c_s^4} - \frac{\boldsymbol{u}_{\mathrm{f}} \cdot \boldsymbol{u}_{\mathrm{f}}}{2c_s^2}\right), \tag{5.13}$$

with the weights w_i that are derived from a Gauss-Hermite quadrature rule. These weights remain constant for the selected velocity set. The lattice speed of sound c_s is also constant and depends on the chosen velocity set.

Using the particle distributions mentioned above, it is also possible to derive macroscopic quantities, such as the fluid density $\rho_{\mathrm{f}}(\boldsymbol{x},t) = \sum_i f_i(\boldsymbol{x},t)$ and momentum $\rho_{\mathrm{f}}\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x},t) = \sum_i \boldsymbol{c}_i f_i(\boldsymbol{x},t)$.

All studies in this paper were performed using the LBM implemented in the open source software OpenLB [75, 94].

5.3.2 Homogenized lattice Boltzmann method

The proposed scheme can be applied to various methods, including PSM. However, in this work we focus specifically on its application to the HLBM introduced by Krause et al. [66]. To allow coupling between components, a continuous model parameter, a confined permeability, $B(\boldsymbol{x},t) \in [0,1]$ is mapped to the entire computational domain [62, 69, 129]. The signed distance to the particle surface is used as a parameter to integrate any shape in this level set function [73].

To account for the influence of the particles on the fluid, we use an exact difference method (EDM) introduced by Kupershtokh et al. [95], as its adaptation has been reported to be superior [69]. It introduces the following source term in Eq. (5.10)

$$S_i(\boldsymbol{x},t) = f_i^{\mathrm{eq}}(\rho_{\mathrm{f}}, \boldsymbol{u}_{\mathrm{f}} + \Delta\boldsymbol{u}_{\mathrm{f}}) - f_i^{\mathrm{eq}}(\rho_{\mathrm{f}}, \boldsymbol{u}_{\mathrm{f}}). \tag{5.14}$$

The required velocity difference $\Delta\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x},t)$ is computed by a convex combination of the fluid and particle velocities [69]

$$\Delta\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x},t) = (B(\boldsymbol{x},t) - 1)\left(\boldsymbol{u}_{\mathrm{p}}(\boldsymbol{x},t) - \boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x},t)\right). \tag{5.15}$$

To account for the influence of the fluid on the particles, the hydrodynamic force is calculated using the momentum exchange algorithm (MEA) of Wen et al. [96]. The local force is then

$$\boldsymbol{F}_{\mathrm{h}}(\boldsymbol{x},t) = \sum_i (\boldsymbol{c}_i - \boldsymbol{u}_{\mathrm{p}}(\boldsymbol{x},t)) f_i(\boldsymbol{x}+\boldsymbol{c}_i,t) + (\boldsymbol{c}_{\bar{i}} + \boldsymbol{u}_{\mathrm{p}}(\boldsymbol{x},t)) f_{\bar{i}}(\boldsymbol{x},t). \tag{5.16}$$

The sum of all local forces of nodes inside the particle, denoted by the position $\boldsymbol{x}_{\mathrm{b}}$, gives the total hydrodynamic force

$$\boldsymbol{F}_{\mathrm{p}}(t) = \sum_{\boldsymbol{x}_{\mathrm{b}}} \boldsymbol{F}_{\mathrm{h}}(\boldsymbol{x}_{\mathrm{b}},t). \tag{5.17}$$

Similarly, the torque is defined as the sum of the cross products between the displacement vectors and the local hydrodynamic forces

$$\boldsymbol{T}_{\mathrm{p}}(t) = \sum_{\boldsymbol{x}_{\mathrm{b}}} (\boldsymbol{x}_{\mathrm{b}} - \boldsymbol{X}_{\mathrm{p}}) \times \boldsymbol{F}_{\mathrm{h}}(\boldsymbol{x}_{\mathrm{b}},t). \tag{5.18}$$

While $\boldsymbol{x}_{\mathrm{b}}$ resides within the particle, the formulation in Eq. (5.16) represents momentum exchange with the surrounding nodes. Consequently, this approach inherently accounts for the influence of the immediate fluid layer surrounding the particle, which is essential to accurately capture fluid-particle interactions.

5.3.3 Discrete contacts

The original discrete contact model [71] consists of three simple mesh-based steps: rough contact detection, its correction, and the calculation of the resulting forces. These steps are briefly explained in Section 5.3.3.1, while Section 5.3.3.2 introduces a method for considering periodic boundaries.

5.3.3.1 Overview

During the coupling of the particles to the fluid, a rough contact detection is conducted. At each lattice node, the signed distance of the particles is evaluated. If the signed distance of two particles at the node is less than half the diagonal of a cell ($d_{\mathrm{s}} < \sqrt{0.75}\Delta x$), these particles are considered to overlap at that particular node. This initial evaluation forms an approximate cuboid that encloses the contact region, a bounding box. This bounding box defines the domain over which iterative calculations are performed to determine, for example, the overlap volume, indentation depth, and contact normal.

However, due to the relatively large lattice spacing, the overlap is not adequately resolved, resulting in an imprecise bounding box. To address this, a subsequent correction step is performed to refine the bounding box and bring it closer to the actual overlap. This correction process involves iterating over the surface of the initial approximation using a predefined number of points in each spatial direction. The number of points is called the contact resolution N_c. At each point, the distance to the actual contact is calculated in discrete directions and this new information is used to improve the accuracy of the bounding box.

In the final step, the contact resolution N_c is used again. Now, it is applied to iterate over the entire overlap region to determine the overlap volume, the contact point, the contact normal, the indentation depth, and the contact force.

For further details on the algorithm, the interested reader is referred to the corresponding publication [71].

5.3.3.2 Periodic boundaries

Typically, a particle is duplicated at periodic particle boundaries, so that it exists on either side of the periodic boundary [69, 130]. However, only the parts that actually intersect the computational domain on each side must be considered.

For contact treatment, we use a similar approach and duplicate particles that intersect the periodic boundary to detect contacts on either side. In case of a contact spanning both sides of the periodic boundary, however, the bounding box would encompass nearly the entire domain, which would lead to inaccurate results. We therefore fix the contact to one side of the periodic boundary. In this work, we define that the contact is always on the side where the center of mass of the particle with the lower ID intersects the domain, which will hereinafter serve as the reference point. Consequently, the contact detection on this side remains unchanged, while it is shifted towards this reference on the opposite side.

These adjustments affect the rough contact detection and the treatment of particle-particle interactions. For the former, the minimum and maximum coordinates of the contact bounding box are fixed to a single side of the periodic boundaries, as described above. If the center of mass of the particle with the higher ID is on the opposite side, it has to be moved to the side of the particle with the lower ID when treating particle-particle interactions. After the contact forces are determined, the particle is moved back to its original position.

It is important to note that the specific definition of the reference point is used in the context of this work. It may be defined differently, of course.

5.4 Parallelization strategy

Given the computational expense associated with the bounding box correction and contact force calculation introduced in Section 5.3.3.1, we propose a parallelization strategy to efficiently distribute the workload. This allows to increase the performance and, thus, the applicability of the discrete contact model.

All of the methods presented are readily available through the open source software OpenLB, starting with version 1.7.0 [75], and in the public repository [1].

5.4.1 Definitions

For clarity and consistency, we provide the following definitions that will be utilized throughout the sections below [73]:

Responsibility The concept of responsibility entails a block or its associated processing unit (PU) assumes the task of managing a particle. This responsibility includes solving the equations of motion and potentially reassigning the particle to another block or PU.

Neighborhood The neighborhood consists of blocks around the block of interest and is defined by a maximum distance from the latter. In this work, the maximum distance corresponds to the largest circumferential radius of all particles in the simulation. The neighboring processes are responsible for the blocks within this neighborhood.

5.4.2 Background

A fundamental aspect of optimizing LBM simulations is the effective distribution of computational tasks across multiple processing units. In practice, a block-based approach is used to achieve this [90, 97]. Here, the computational domain is decomposed into a collection of distinct blocks. Each block is assigned to a specific processing unit to enable simultaneous execution of computations. A key feature of this approach is that the processing units operate primarily on local data and data exchange is limited to the boundaries of adjacent blocks, where information is selectively shared to ensure coherent execution of the simulation.

[1] https://gitlab.com/openlb/release

Marquardt et al. [73] propose a similar scheme for the decomposition of surface resolved particles. In this scheme, the assignment of a particle to a particular processing unit is intricately linked to the position of the center of mass of the particle. The processing unit responsible for the block containing the center of mass assumes responsibility for the particle. In the likely scenario that the particle's surface overlaps another block, however, the responsible process has no access to the fluid data required for coupling. To consider such scenarios, the processing unit responsible for the neighboring block that intersects the particle surface calculates the partial surface force and communicates it to the processing unit responsible for the particle in question, where the sum of all partial surface forces is used to solve the equations of motion. To account for the dynamic nature of simulations, particle assignment is revised after the particle's position has been updated. This updated assignment is communicated, along with the particle's data, to all processing units responsible for blocks in the particle's vicinity.

The previous parallelization strategy of the discrete contact model is incompatible with the particle decomposition scheme. This incompatibility stems from the requirement that each PU must have information about all particles in order to solve each contact independently. This highlights another problem with the previous approach: it becomes very inefficient as the number of contacts grows, which in turn makes the studies targeted in this work infeasible.

5.4.3 Distribution of contacts

In order to distribute the computational effort of the contact treatment, we propose a simple assignment of contacts to processes in a communication-ideal strategy [97].

For particle-wall contacts, identifying the responsible PU is trivial, because the non-moving walls are known to all PUs. The unit responsible for the particle is also responsible for the contact treatment, which limits the number of communications.

However, particle-particle contacts are more complex due to their dynamic nature and the local storage of data. Therefore, we categorize the contacts based on the available data. For this we refer to Figure 5.1. Each of these figures shows four distinct blocks, each corresponding to a different PU, as indicated by the numbers in the corners. In addition, each figure shows the surface of two particles as solid lines along with their respective centers of mass as pluses.

For simplicity, we assume that the PUs only know the particle data when

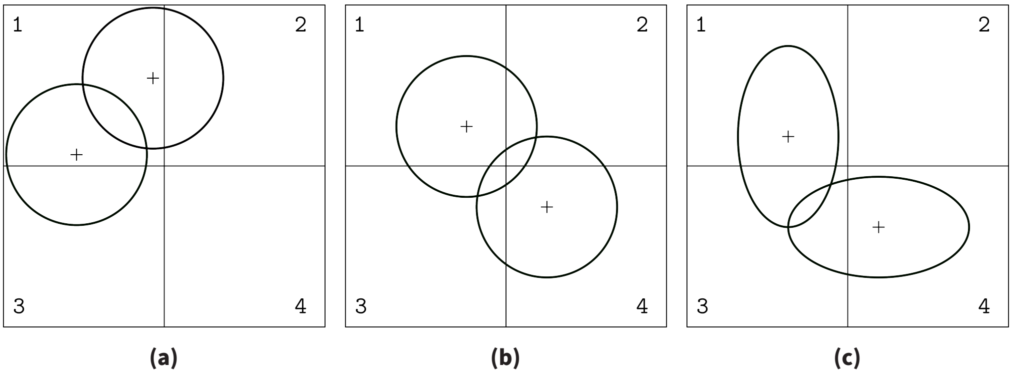


Figure 5.1: *Illustration of three types of particle-particle contacts: The centers of mass of both particles are in the same block (a), the centers of mass of both particles are in different blocks, but both particles are known to at least one of the responsible PUs (b), and none of the responsible PUs knows both particles, only a third PU knows all data (c).*

the surface intersects the corresponding block. As the particle decomposition scheme [73] aims to avoid unnecessary complexity, however, the actual algorithm does not check for intersections and communicates the particle data to all neighbors instead.

In Figure 5.1(a), we illustrate contacts between particles that share the same responsible PU, identified by the number 1 in the example. In this scenario, the unit assumes responsibility for handling the contact and no intermediate communication is required. Moving on to Figure 5.1(b), we see that the particles involved have different responsible PUs, but at least one of these units stores the data of both particles, as their surfaces intersect the corresponding block. If both particles are known to both responsible PUs, the unit with the lower ID is designated to be responsible for processing the contact. In this example, PU 1 would assume responsibility. Finally, as shown in Figure 5.1(c), none of the PUs responsible for the particles has information about both particles. In this scenario, we assign responsibility for handling the contact to the PU with the lowest ID that has access to the most recent data for both particles. In the example shown, this would be PU 3.

Given the types described above, we identify four additional steps as outlined below.

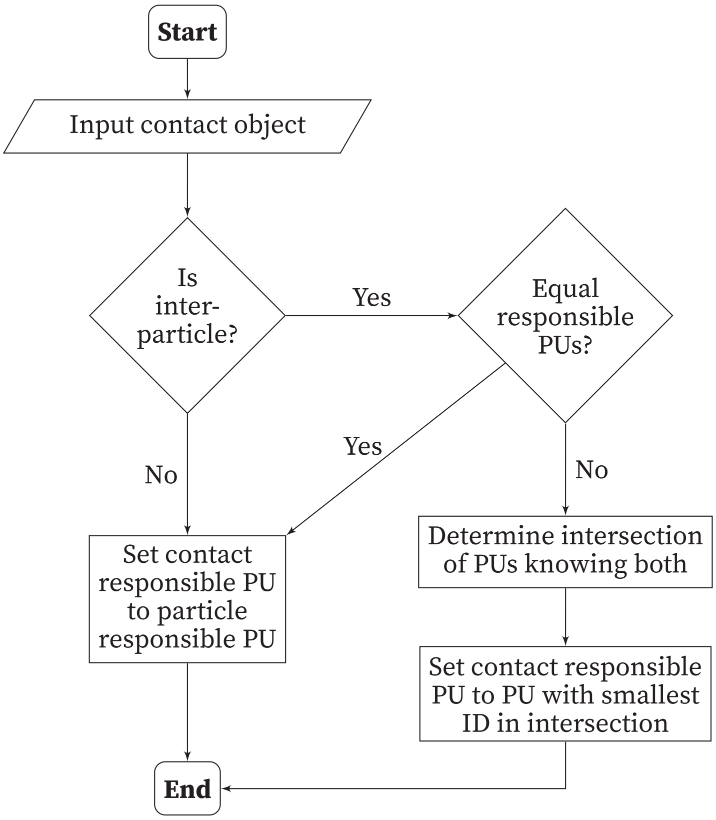

Start
Input contact object
Is inter-particle?
Yes
Equal responsible PUs?
No
Yes
No
Set contact responsible PU to particle responsible PU
Determine intersection of PUs knowing both
Set contact responsible PU to PU with smallest ID in intersection
End

Figure 5.2: *Visualization of the algorithm to determine a contact responsible PU.*

5.4.3.1 Determination of the responsible processing units

For all processes involving potential interactions between two particles or between a particle and a wall, sufficient information is available to determine the responsible PU, as described above. This information includes the knowledge of the exact position of the particle and of the block it is located in as well as the maximum circumferential radius of the entire particle collective. Consequently, the task of identifying the responsible PU is performed within the local context of the given process, following the algorithm shown in Figure 5.2.

5.4.3.2 Communication of the detected contacts

Rough on-lattice contact detection, as described in [71], serves to consolidate all contact components. We deliberately refrain from performing the rough contact detection for already established contacts in order to prevent its errors

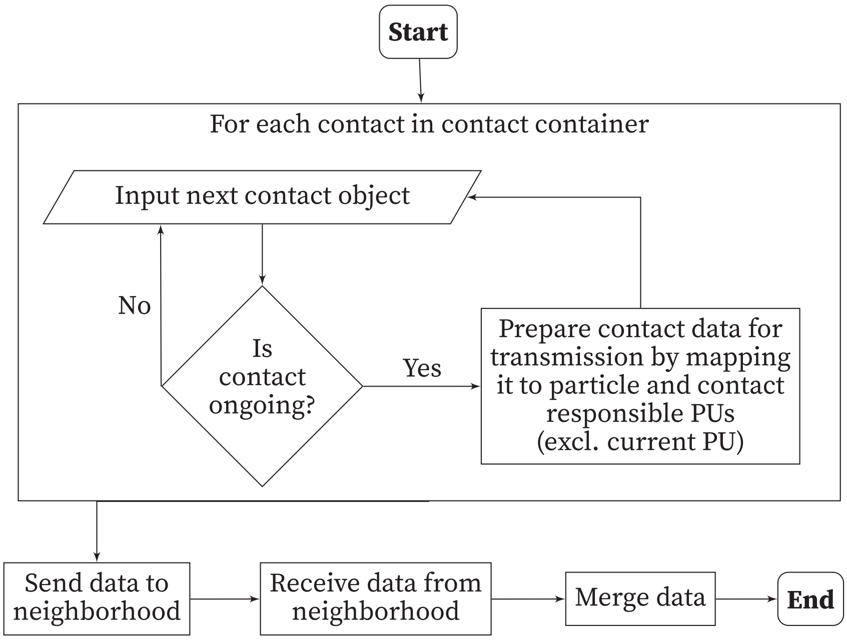

Start
For each contact in contact container
Input next contact object
No
Is contact ongoing?
Yes
Prepare contact data for transmission by mapping it to particle and contact responsible PUs (excl. current PU)
Send data to neighborhood
Receive data from neighborhood
Merge data
End

Figure 5.3: *Basic procedure for communicating detected contacts. Note that this step takes place after the determination of the contact responsible PU (cf. Section 5.4.3.1), during which the destination PUs for this step may be determined to save computational resources.*

from being reintroduced.

However, we still communicate these pre-existing contacts to ensure that responsible processes remain informed of any contacts that may have gone undetected on their side. To accomplish this, each process having knowledge of an ongoing contact sends the relevant data to processes responsible for managing at least one particle or handling the contact. Neighboring processes not involved in particle or contact management receive empty requests.

The data is only communicated if the objects still overlap, see Figure 5.3, and includes critical details such as global particle IDs or global particle and wall IDs, the minimum and maximum bounding box coordinates, the damping factor, and the pre-evaluated responsible PU.

The receiving PU integrates the transmitted data with any existing data to ensure that the collective sub-bounding boxes accurately define the bounding box that encompasses the entire contact. This integration involves determining the combined bounding box by selecting the minimum coordinates from all of the sub-bounding boxes for the lower boundaries, as well as the maximum coordinates. The damping factor initially has an invalid negative value. Consequently, the contacts either have identical values or have positive and negative discrepancies. During the merge, the valid positive value takes precedence.

5.4.3.3 Communication of the contact force and torque

As described above, the PU that determines the contact force and the corresponding torque may not be responsible for solving the equations of motion, but the data are required for solving these equations. The PU responsible for the contact sends the resulting force and torque, along with the global particle ID, to each PU responsible for at least one of the particles in contact. Other processes in the neighborhood that do not meet these conditions receive an empty request.

To reduce the computational effort, this step is combined with the contact force and torque calculation as shown in Figure 5.4, which gives details of the underlying procedure.

The PUs responsible for solving the particles receive the results of the contact treatment and add them to the existing forces and torques so that the sum can be used to solve the equations of motion.

5.4.3.4 Communication of the contacts after the contact treatment

Awareness of an existing contact and its latest data is limited to the responsible PU. In the case of particle-particle contacts, it is therefore imperative that the PU communicates contact information to all processes that hold data of both particles. For particle-wall contacts, data must be shared with all processes that have access to the involved particle's data. If neighboring blocks do not fulfill the above criteria, they receive an empty request. When the latest data is received, it overrides any outdated data that may still be present. Figure 5.5 provides a structured overview of the process.

This ensures that no data is lost and that a consistent damping factor is maintained for the duration of the contact. The contact data mentioned above should include either the two global particle IDs (for particle-particle contacts) or a global particle ID and a global wall ID (for particle-wall contacts). It also includes the damping factor calculated from the initial relative velocity as described in Eq. (5.6).

We choose to also communicate the updated minimum and maximum coordinates of the contact's bounding box. This information helps to achieve

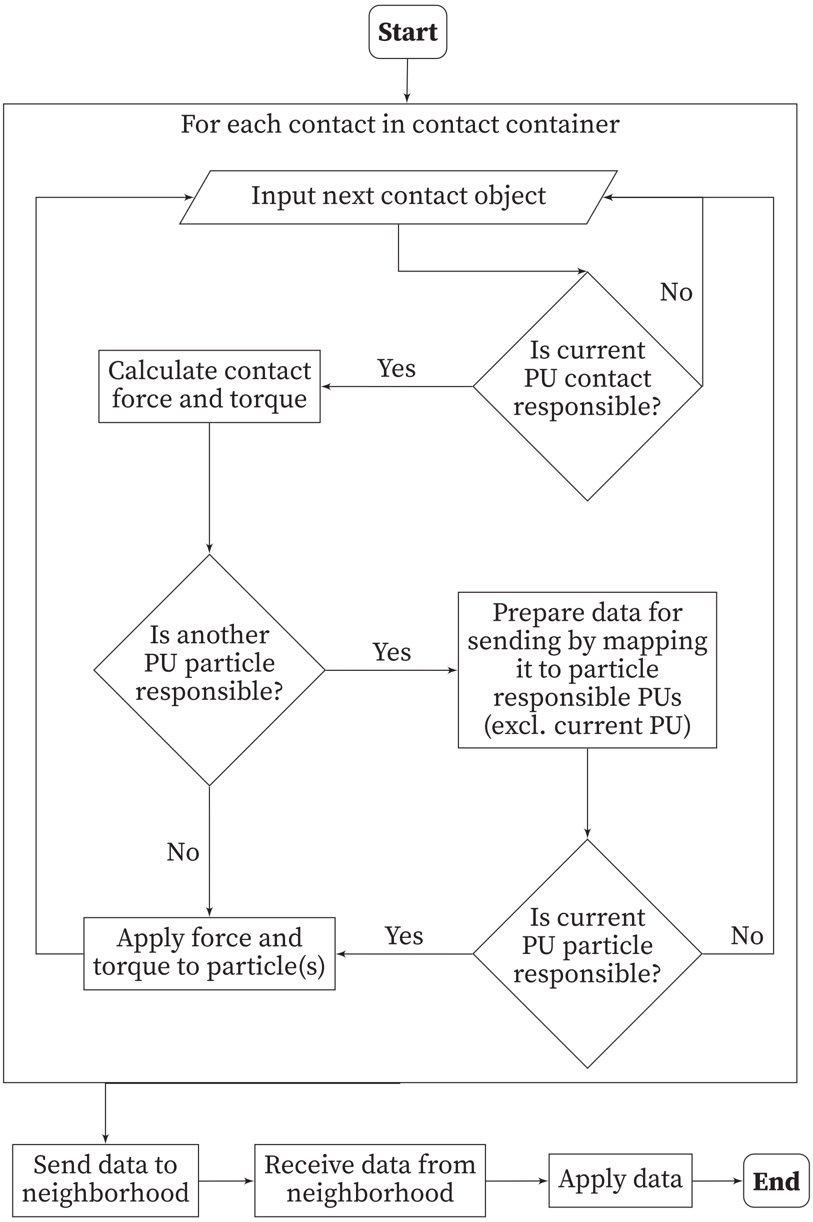

Start
For each contact in contact container
Input next contact object
Is current PU contact responsible?
No
Yes
Calculate contact force and torque
Is another PU particle responsible?
Yes
Prepare data for sending by mapping it to particle responsible PUs (excl. current PU)
No
Is current PU particle responsible?
Yes
No
Apply force and torque to particle(s)
Send data to neighborhood
Receive data from neighborhood
Apply data
End

Figure 5.4: *Visualization of the steps required to communicate the results of the contact treatment.*

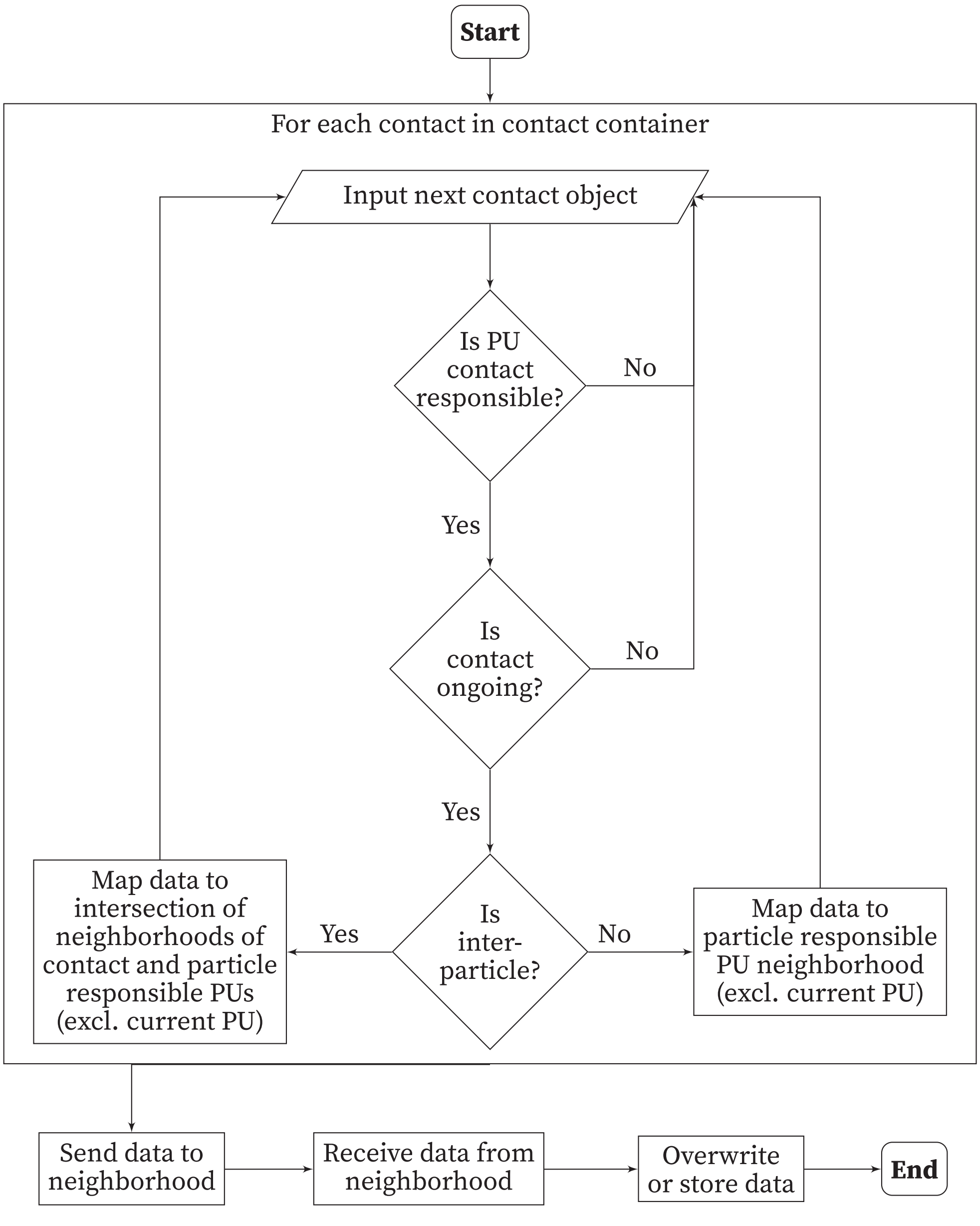

Start
For each contact in contact container
Input next contact object
Is PU contact responsible?
No
Yes
Is contact ongoing?
No
Yes
Is inter-particle?
Yes
No
Map data to intersection of neighborhoods of contact and particle responsible PUs (excl. current PU)
Map data to particle responsible PU neighborhood (excl. current PU)
Send data to neighborhood
Receive data from neighborhood
Overwrite or store data
End

Figure 5.5: *Illustration of the communication of contacts after the contact treatment.*

higher accuracy by minimizing the effects of the rough contact detection. When considering that the particle positions change only slightly between successive time steps, it is inevitable that the bounding box changes only slightly as well. Subsequent application of the bounding box correction yields improved results without reintroducing errors from rough contact detection.

5.4.4 Time step algorithm

To give a structured overview, Algorithm 5.1 outlines the basic procedure. It is worth noting that it includes sub time steps with constant hydrodynamic forces [71]. For the sake of simplicity and to keep performance comparable with and without the contact model, we omit sub time steps in the following sections.

Algorithm 5.1: Basic LBM time step algorithm using PSM with the particle and contact decomposition scheme

```
for all time steps do
    Couple fluid to particles;                                  ▷ Using the MEA
    Communicate surface forces and torques;                         ▷ See [73]
    for all sub time steps do
        Compute contact forces;                                     ▷ See [71]
        Communicate contact forces;                              ▷ See 5.4.3.3
        Apply external forces;                               ▷ Such as gravity
        Solve equations of motion;
        Communicate post-treatment contacts;                     ▷ See 5.4.3.4
        Evaluate particle assignment;                               ▷ See [73]
        Communicate data and assignment;                            ▷ See [73]
        Couple particles to fluid with contact detection;           ▷ See [71]
        Determine responsible PUs;                               ▷ See 5.4.3.1
        Communicate detected contacts;                           ▷ See 5.4.3.2
        Apply periodic boundary to contacts (optional);          ▷ See 5.3.3.2
    end
    Perform collision and streaming;
    Increase time step;
end
```

5.5 Application to hindered settling

As the hindered settling of spherical particles has been studied extensively and many correlations are known, we use these to evaluate the proposed method.

Most correlations describe the ratio of the average settling velocity of the bulk $\bar{u}_\mathrm{p}$ and the single particle settling velocity u^* using a power law approach [98]

$$\frac{\bar{u}_\mathrm{p}}{u^*} = k(1-\phi_\mathrm{p})^n. \tag{5.19}$$

Here, k is a prefactor, n is the expansion index, and ϕ_p is the particle volume fraction. The single particle velocity reads

$$u^* = \sqrt{\frac{4gD_\mathrm{s}}{3C_\mathrm{d}}\left(\frac{\rho_\mathrm{p}-\rho_\mathrm{f}}{\rho_\mathrm{f}}\right)}, \tag{5.20}$$

with the standard gravity $g = 9.80665\ \mathrm{m/s^2}$, the diameter of the spherical particle D_s, the drag coefficient C_d, and the fluid and particle densities ρ_f and ρ_p. In this study, we use the well-known drag coefficient correlation by Schiller and Neumann [99]

$$C_\mathrm{d} = \frac{24}{Re}\left(1+0.15Re^{0.687}\right), \tag{5.21}$$

which is valid for $Re = u^* D_s/\nu < 800$.

Early studies neglected the prefactor k, i.e. $k = 1$, and focused on the expansion index n. Richardson and Zaki [77] propose

$$n = \begin{cases} 4.65 & \text{for } Re < 0.2 \\ 4.35Re^{-0.03} & \text{for } 0.2 \leq Re < 1 \\ 4.45Re^{-0.1} & \text{for } 1 \leq Re < 500 \\ 2.39 & \text{for } 500 \leq Re \end{cases}. \tag{5.22}$$

Using a power law-based approach, Garside and Al-Dibouni [79] suggest

$$n = \frac{5.1+0.27Re^{0.9}}{1+0.1Re^{0.9}}. \tag{5.23}$$

For the latter correlation, superior accuracy has been reported [98]. Note that Richardson and Zaki [77] use a Reynolds number that depends on the velocity of a single particle in an infinite fluid, while Garside and Al-Dibouni [79] consider the single particle settling velocity in the domain under consideration. Due to the chosen simulation setup, however, these originally different Reynolds numbers coincide in the present work.

Later studies [20, 80, 81] suggest including a prefactor $k \in [0.8, 0.9]$. Yao et al. [55] use numerical experiments to derive an Archimedes number-dependent equation with a reported coefficient of determination of 0.86 for $Ar \in [21, 2360]$, which reads

$$k = 0.89 \exp\left(-\frac{Ar}{10^5}\right), \tag{5.24}$$

where the Archimedes number is defined as

$$Ar = \frac{g D_\mathrm{s}^3 \rho_\mathrm{f}(\rho_\mathrm{p} - \rho_\mathrm{f})}{\eta^2}. \tag{5.25}$$

5.5.1 Simulation setup

To perform a numerical study of the scenario described above, we use spherical particles with a diameter of $D_\mathrm{s} = 1.5$ mm, randomly distributed within a cubic domain with an edge length of L on either side. The domain has periodic boundaries in all directions and is completely occupied by a fluid characterized by a density of $\rho_\mathrm{f} = 1000\,\mathrm{kg/m^3}$. An illustration of an example setup at a particle volume fraction of about 30% is provided in Figure 5.6.

Initially, the particles rest, but they experience an acceleration in the z-direction due to the force $F_g = (\rho_\mathrm{f} - \rho_\mathrm{p})V_\mathrm{p} g$, where V_p represents the volume of a single particle. Following previous studies [53, 54, 56, 98], a pressure gradient is applied to prevent unbounded acceleration and infinite velocities so that the suspension has a net volume flow of zero. The Archimedes number takes values between 500 and 2000 and the density ratio is fixed to 3.3, since its influence was found to be negligible [55]. The particle volume fraction ranges from 10% to 30%. In addition, we maintain a consistent lattice relaxation time $\tau = 0.55$ across all simulations, with variations in the number of cells N used to resolve the sphere's diameter in the subsequent simulations.

As suggested by previous studies [71], we use a contact resolution of 8 and increase the particle size during the contact treatment to improve the accuracy in viscous fluids. Therefore, we enlarge the particles by $\Delta x/5$. Additionally, we use a coefficient of restitution $e = 0.926$ and coefficient of kinetic friction $\mu_k = 0.16$ [131]. The coefficient of static friction is set to 0.32. Similar to studies

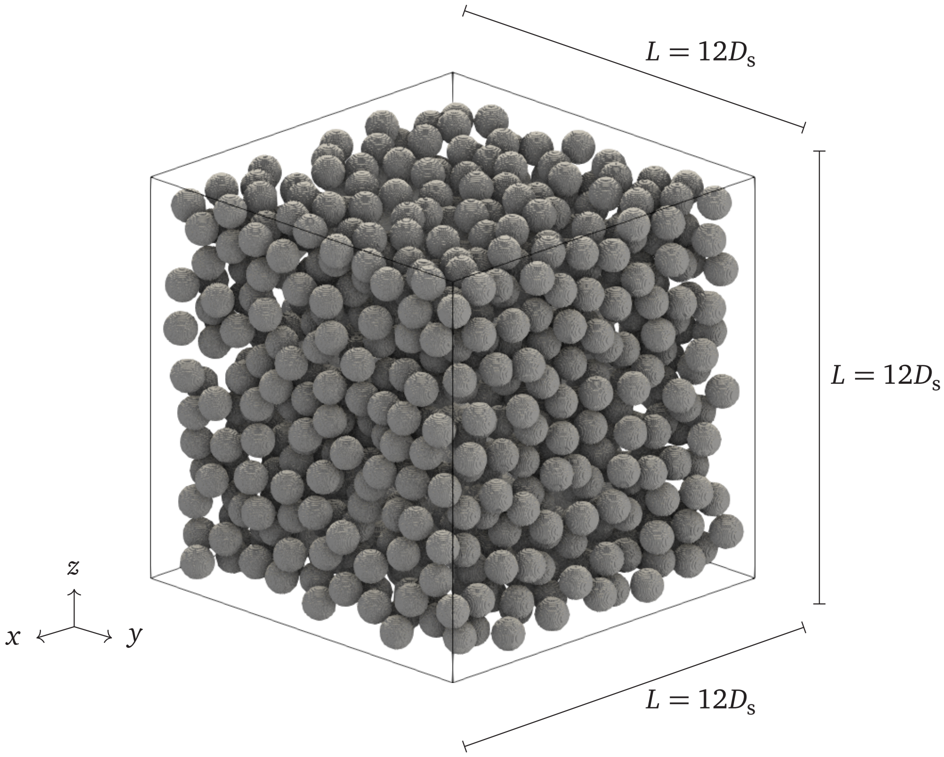


Figure 5.6: *Simulation domain with spherical particles on their preassigned initial positions using a resolution of $N = 27$ cells per sphere diameter and particle volume fraction of about 30%.*

in literature [53, 54], we significantly reduce the Young's modulus to $E = 5$ kPa to ensure stable simulations. This adjustment is necessary, because LBM simulations are constrained by a minimum lattice relaxation time $\tau > 0.5$ [90], which in turn limits the smallest feasible time steps Δt for the chosen resolution. However, the proposed configuration becomes increasingly susceptible to instabilities for $\tau < 0.55$. This tendency is likely due to the emergence of small channels between particles, potentially leading to increased local velocities and a decrease in the maximum permissible local lattice velocity for $\tau < 0.55$ [90]. Fortunately, the findings presented in Section 5.5.3 suggest that the effect of the softening is sufficiently small, as a good agreement with the correlations is visible.

The simulations cover a time of $400t^*$, with the normalized time $t^* = tD_s g(\rho_p - \rho_f)/(18\eta)$. In all simulations, averaging begins after $50t^*$. The data of

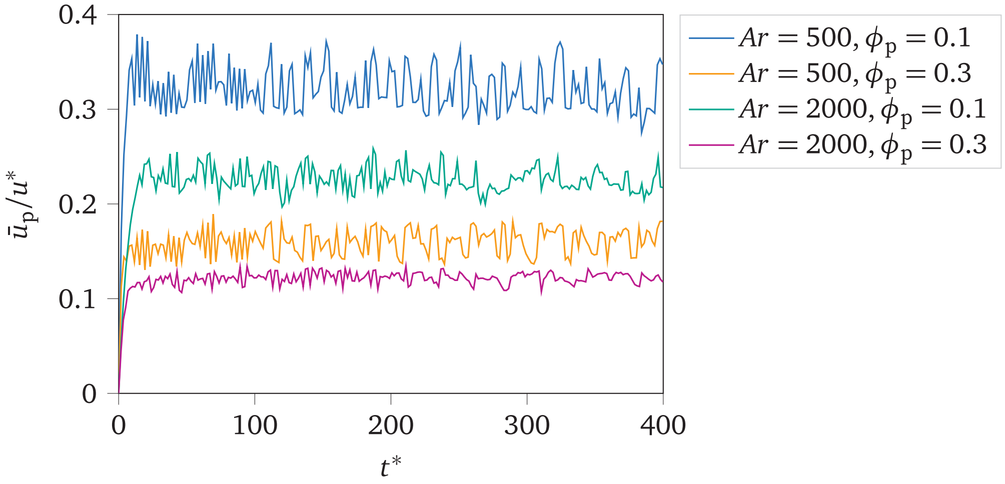


Figure 5.7: *Normalized average settling velocity versus the normalized time t^*.*

simulations using $L = 12D_s$ and $N = 18$ shown in Figure 5.7 illustrate that this is sufficient. Figure 5.7 shows the normalized average settling velocity $\bar{u}_\mathrm{p}/u^*$ over the normalized time t^* for the minimum and maximum Archimedes number $Ar \in \{500, 1000\}$ as well as minimum and maximum particle volume fraction $\phi_\mathrm{p} \in \{0.1, 0.3\}$ considered. Lower ratios are obtained when dealing with higher particle volume fractions and lower Archimedes numbers. Additionally, the ratios fluctuate, which is more pronounced for smaller Archimedes numbers and particle volume fractions. For all cases, the average value is reached well before $t^* = 50$. Afterwards, only oscillations around that value occur. Thus, the aforementioned averaging start time is adequate for the extreme cases and for all intermediate parameters.

5.5.2 Grid independence study

We perform a grid independence study by considering a particle volume fraction of 15% at an Archimedes number of 1000 in a domain with an edge length of $L = 12D_s$ and the resolutions $N \in \{5, 7, 9, 12, 15, 18, 27\}$. To confirm grid independence, we compare the above resolutions with the baseline resolution of $N = 27$. The results are shown in Figure 5.8. This figure shows the relative error, calculated using the L^2 norm as described in [90], plotted against the different grid resolutions used. Results without the contact model are indicated by orange pluses, while results with the contact model are marked by blue

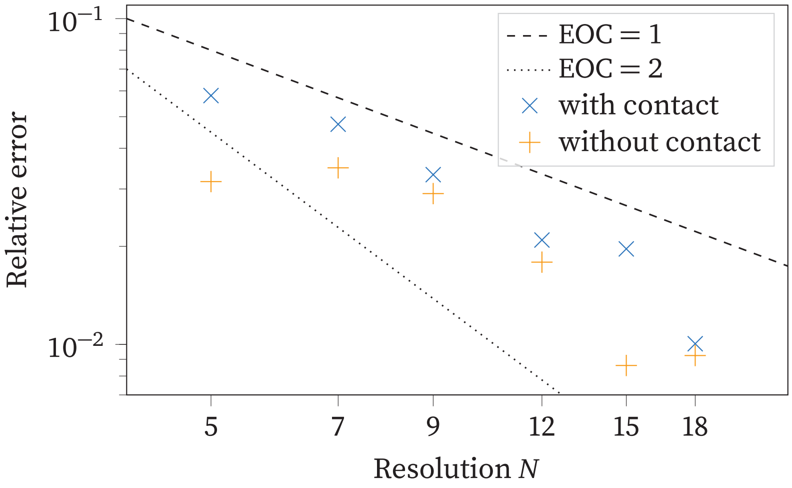


Figure 5.8: *Relative error with and without the contact model in L^2 norm versus the resolution of the sphere's diameter N. The references use the resolution $N = 27$.*

crosses. The figure also includes lines representing the experimental orders of convergence (EOC) with values of both 1 and 2.

In Figure 5.8, the error decreases in both cases as the resolution increases. It is remarkable that for most resolutions, the relative errors with and without explicit contact treatment have similar values, except for the N values of 5 and 15. These outliers make it more difficult to assess which EOC line the relative errors without contact modeling adhere to, but their evolution appears to be closest to an EOC of 1. In contrast to this, the relative errors with contact treatment clearly follow the line of EOC $= 1$. This suggests that in both cases, the error is roughly halved when the resolution is doubled. It is important to note that in both scenarios, the relative error falls below 3% for $N \geq 12$. For $N = 18$ it is close to 1%. For this reason and because of the small time steps needed for the contact treatment, we choose $N \geq 18$ for the following studies.

5.5.3 Validation

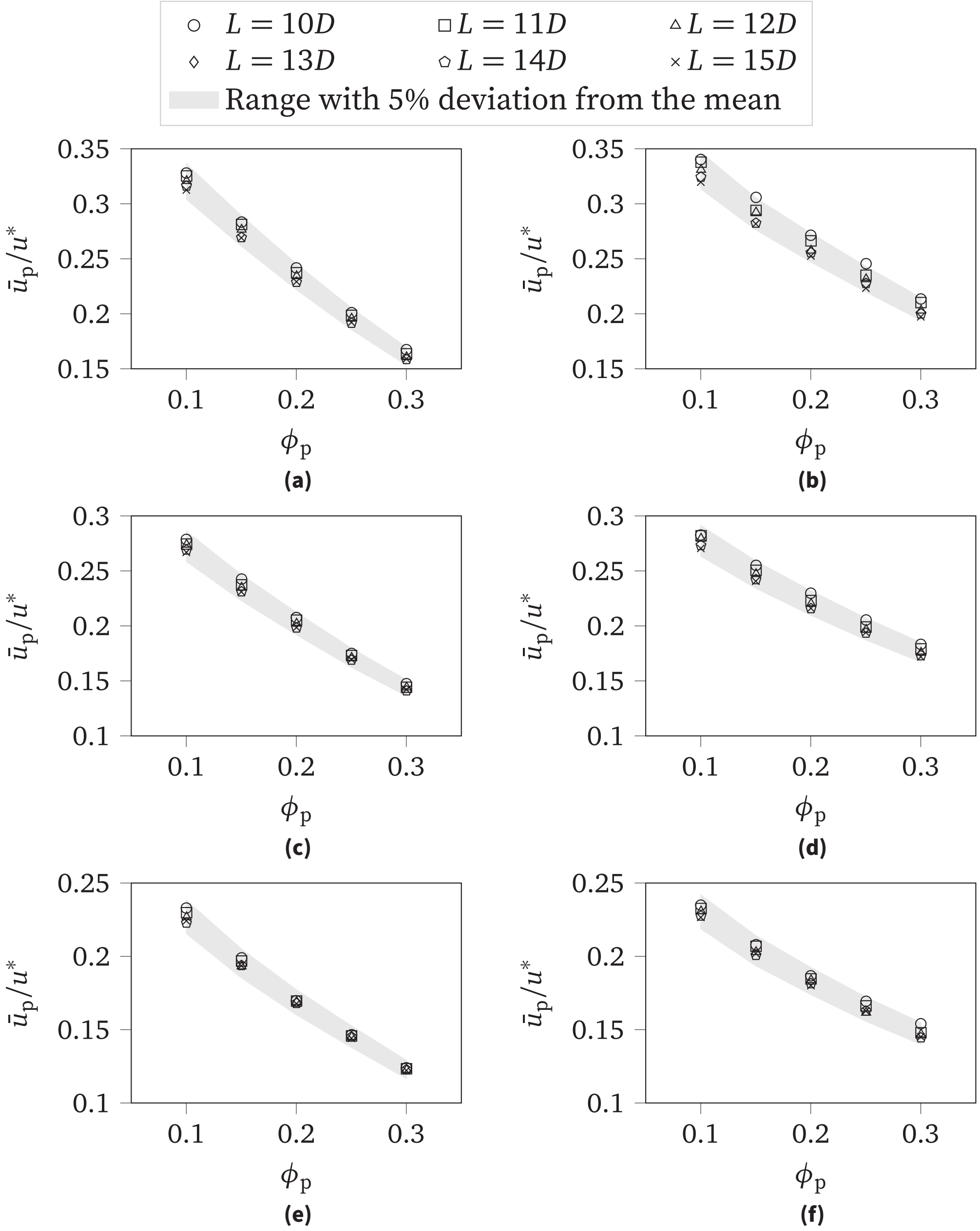
The following validation consists of two parts. First, we aim to verify the used setup by employing various simulations with different Archimedes numbers Ar, edge lengths L, and initial particle positions. Figure 5.9 shows the explicit parameters of the different setups and plots the ratio of average settling velocity and single particle settling velocity versus particle volume fraction. On the left, an explicit contact treatment is used, on the right, it is not. The top plots correspond to simulations conducted with $Ar = 500$, the middle plots with $Ar = 1000$, and the bottom plots with $Ar = 2000$. This figure also displays the range with a 5% deviation from the mean at each particle volume fraction as a light gray area.

In Figure 5.9, all data points lie in the highlighted area, indicating that the difference from the mean is less than 5% in all simulations. For the simulations without explicit contact treatment and at $Ar = 500$, however, there is one outlier at a particle volume fraction of 25% and an edge length of $L = 10\ D_s$. Furthermore, we see that for low Archimedes numbers the data points for similar particle volume fractions are further apart than for higher Ar. This is particularly significant when an explicit contact treatment is used. A similar trend is seen when looking at the results obtained with a contact treatment and increasing particle volume fractions. The higher the particle volume fractions are, the narrower are the data. Without an explicit contact model, the size of the data range at each particle volume fraction appears to remain approximately the same.

The differences in the trend over particle volume fraction are likely due to the considerably higher error at high particle volume fractions without an explicit model for particle-particle contacts. This implies that the contact model is necessary at high particle volume fractions. Furthermore, the above observations suggest that in the range of edge lengths considered, their effect is sufficiently small. Similarly, the effect of the initial random positions is negligible. However, an outlier is present at $L = 10D_s$ and since a higher number of particles is of interest to later performance evaluations, we only consider edge lengths $L \geq 12D_s$ from now on.

The second part of the validation focuses on comparing the results of simulations using $L = 12D_s$ with the above correlations. For visual comparison, we plot the ratio of the average settling velocity and the single particle settling velocity against the particle volume fraction in Figure 5.10. The results with and without an explicit contact model are shown as blue crosses and orange pluses, respectively. We also add the original correlation of Richardson and Zaki [77], i.e. Eq. (5.19) with Eq. (5.22) and $k = 1$ as a dashed green line. The red solid line shows the results for the correlation of Garside and Al-Dibouni [79] using the prefactor k as suggested by Yao et al. [55], i.e. Eq. (5.19) with Eq. (5.23) and Eq. (5.24). The top plot corresponds to $Ar = 500$, the middle to $Ar = 1000$, and the bottom to $Ar = 2000$.

In all of these plots, we see that the average bulk settling velocity decreases with increasing particle volume fraction for all simulations and correlations.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.9: *Visualization of the domain dependence of the results by plotting the normalized mean sedimentation velocity versus the particle volume fraction ϕ_p for different domain sizes. On the left, the simulations were performed with an explicit contact model and on the right without. The top plots consider $Ar = 500$, the middle plots $Ar = 1000$, and the bottom plots $Ar = 2000$.*

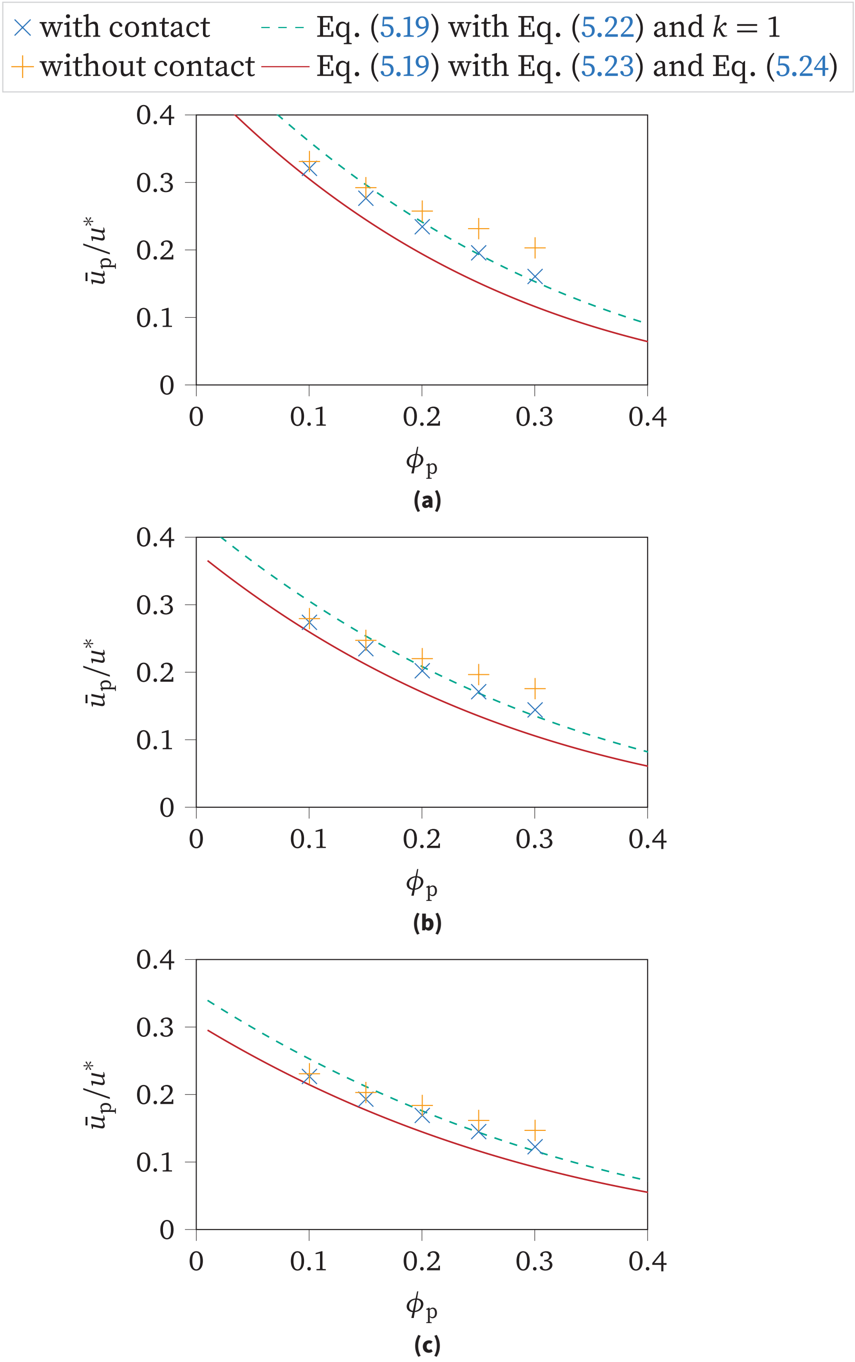
Eq. (5.19) with Eq. (5.23) and Eq. (5.24) (solid red line), however, predicts lower velocities than Eq. (5.19) with Eq. (5.22) and $k = 1$ (dashed green line). While the simulation results are close to the former for low particle volume fractions, the simulation discrepancy increases for higher volume fractions. Thus, the simulations with a contact treatment are close to the predictions of Richardson and Zaki [77], while the simulations without an explicit contact treatment exhibit a notable overestimation. This overestimation is stronger for lower Archimedes numbers and higher particle volume fractions. The latter also has a significant effect on the difference between the simulated average settling velocities with and without the contact model, as they are very similar at low volume fractions and then diverge as they increase.

The increase in the difference between the simulation results is explained by the fact that at low particle volume fractions there are far less contacts and therefore they have a smaller and even negligible influence. However, as the particle volume fraction increases, there are more and longer contacts, which increases their influence. When these contacts are not modeled, a clear error is observed. This error becomes smaller for higher Archimedes numbers, probably due to the increased importance of hydrodynamic forces in these scenarios.

The trend of the simulation results with the explicit contact model also indicates an overestimation at large particle volume fractions. There are three main reasons. First, we have adjusted the Young's modulus to obtain more stable simulations and, thus, introduced an error especially at large particle volume fractions, although this error is limited and a good agreement is obtained. Second, the enlargement of the particles during the contact treatment is likely to introduce an error. Replacing it with a dedicated lubrication force model may improve the results. Third, although the settling velocities are lower than for single particle settling, the fluid velocities increase at large particle volume fractions, especially in the small channels. This potentially leads to high local lattice velocities exceeding the 0.01 and 0.04 accuracy limits observed in previous studies using HLBM [69].

5.5.4 Performance

The goal of this section is to delve deeper into the performance analysis to quantitatively evaluate the feasibility and the impact of the contact treatment versus no treatment, by comparing the million lattice site updates per second (MLUPs). We exclude the former parallelization strategy from our analysis



(a)

(b)

(c)

Figure 5.10: *Comparison of simulation results with and without the use of an explicit contact model with correlations from literature. (a)* $Ar = 500$*, (b)* $Ar = 1000$*, and (c)* $Ar = 2000$.

due to its limitations, described in Section 5.4.2, which make the simulations infeasible. For the numerical experiments, we use Intel Xeon Platinum 8368 CPUs, and each node is equipped with 76 CPU cores.

Figure 5.11 shows a compact overview of 288 separate simulations in four different performance plots of MLUPs versus the number of utilized nodes in a range from 1 to 12. In the top plots, the simulation domain has an edge length of $L = 12D_s$ and in the bottom plots of $L = 15D_s$. The left plots use a resolution of $N = 18$ and the right plots use $N = 27$. In all cases, three different particle volume fractions, i.e. 10%, 20%, and 30%, are considered with (solid lines) and without (dashed lines) consideration of contacts. Note that $L = 15D_s$ and $\phi_\mathrm{p} \approx 0.3$ give a total of 1934 particles, which is the maximum number of particles considered. Conversely, the minimum number of particles is 191, which occurs when $L = 12D_s$ and $\phi_\mathrm{p} \approx 0.1$.

The following observations result: First, the more particles are considered, the lower are the MLUPs. Second, smaller resolutions contribute to a reduction in MLUPs. Third, as the domain size decreases, the MLUPs tend to decrease as well. Finally, considering contacts during the simulation further reduces the MLUPs. For the latter, note that at low resolutions and problem sizes, see Figure 5.11(a), the impact is quite significant, as the MLUPs can decrease by a factor of about 3. However, increasing the resolution, see Figure 5.11(b), or the domain size, see Figure 5.11(c), decreases the impact of the contact treatment. It is almost negligible for large problem sizes, especially for small particle volume fractions, see Figure 5.11(d).

The above observations reveal a noticeable impact of the contact treatment on the computational performance. As mentioned above, however, this trade-off is necessary, especially at high particle volume fractions, to correctly capture the physics of particle flows. Furthermore, the performance of the four-way coupled simulations shows that it is now feasible to consider thousands of surface resolved particles thanks to the proposed parallelization strategy.

5.5.5 Spheres versus cubes

This section compares the average settling velocities of spheres and cubes. These results are of preliminary character and are intended to demonstrate the application to complex shapes. In particular, we consider cubes, because they introduce complexity in addition to being non-spherical in terms of edges and corners. The simulation setup for the volume-equivalent cubes follows that of the spheres described in Section 5.5.1. Note that the initial positions of the

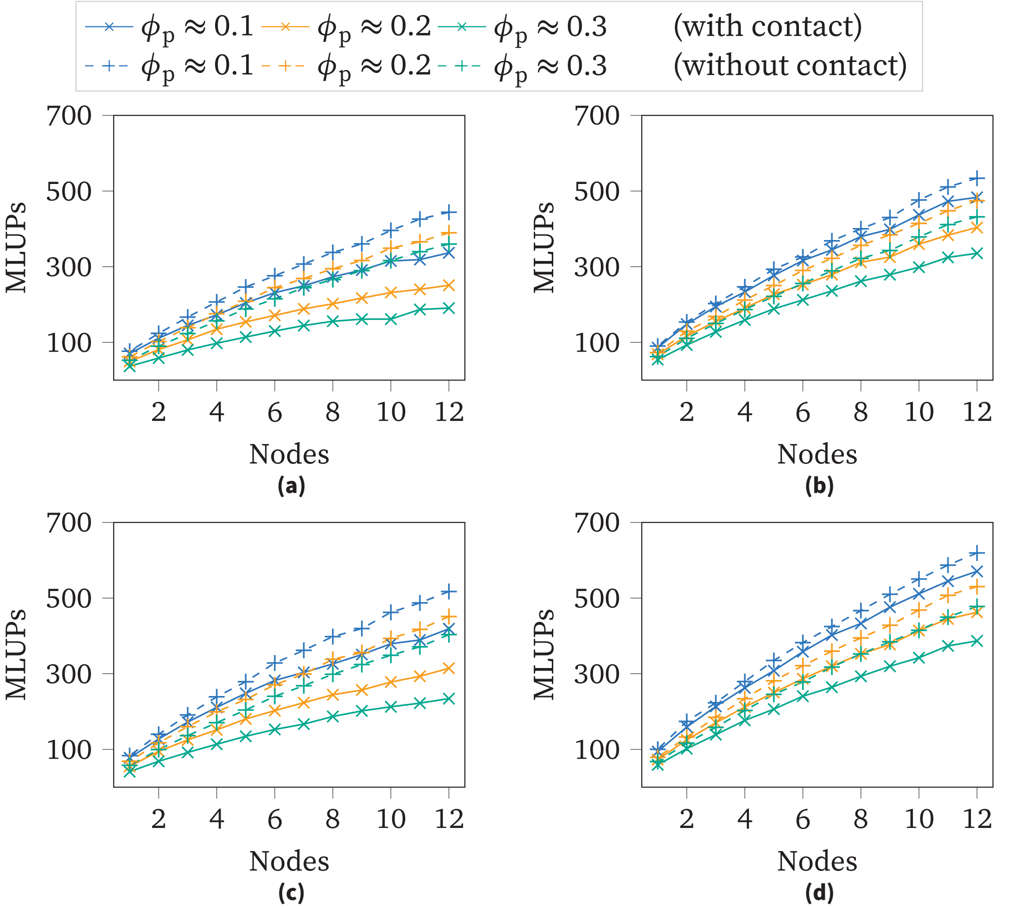


Figure 5.11: *Comparison of the MLUPs versus the number of nodes used for simulations with and without explicit contact treatment for different particle volume fractions. At the top, the simulations with an edge length of $L = 12D_s$ are shown. At the bottom, the edge length is $L = 15D_s$. On the left, a resolution of $N = 18$ and on the right of $N = 27$ is used.*

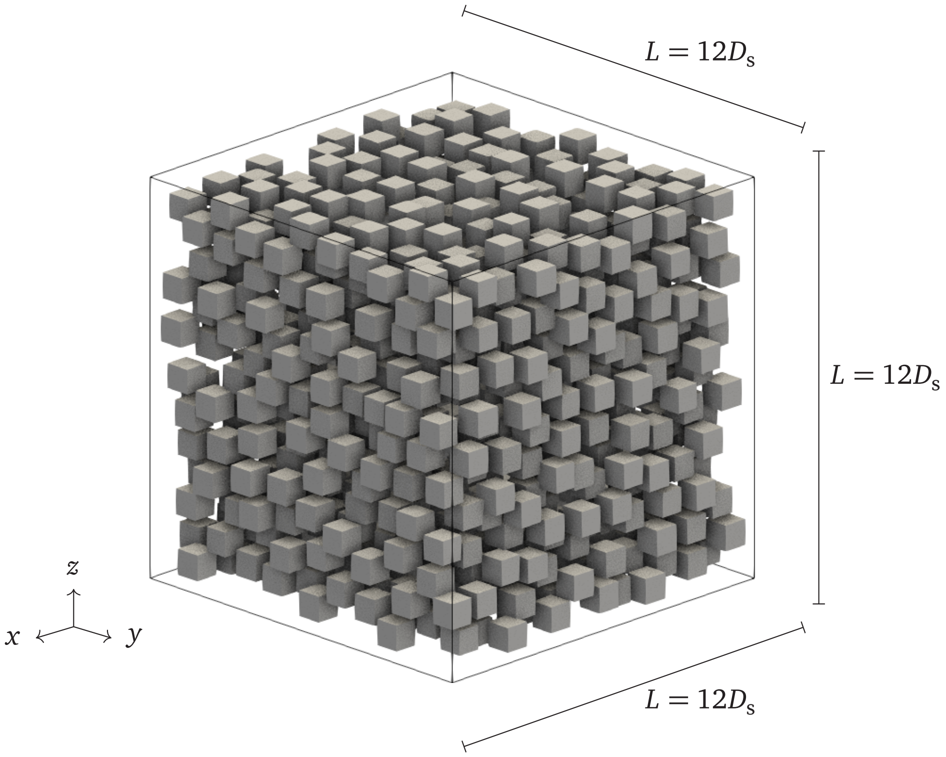
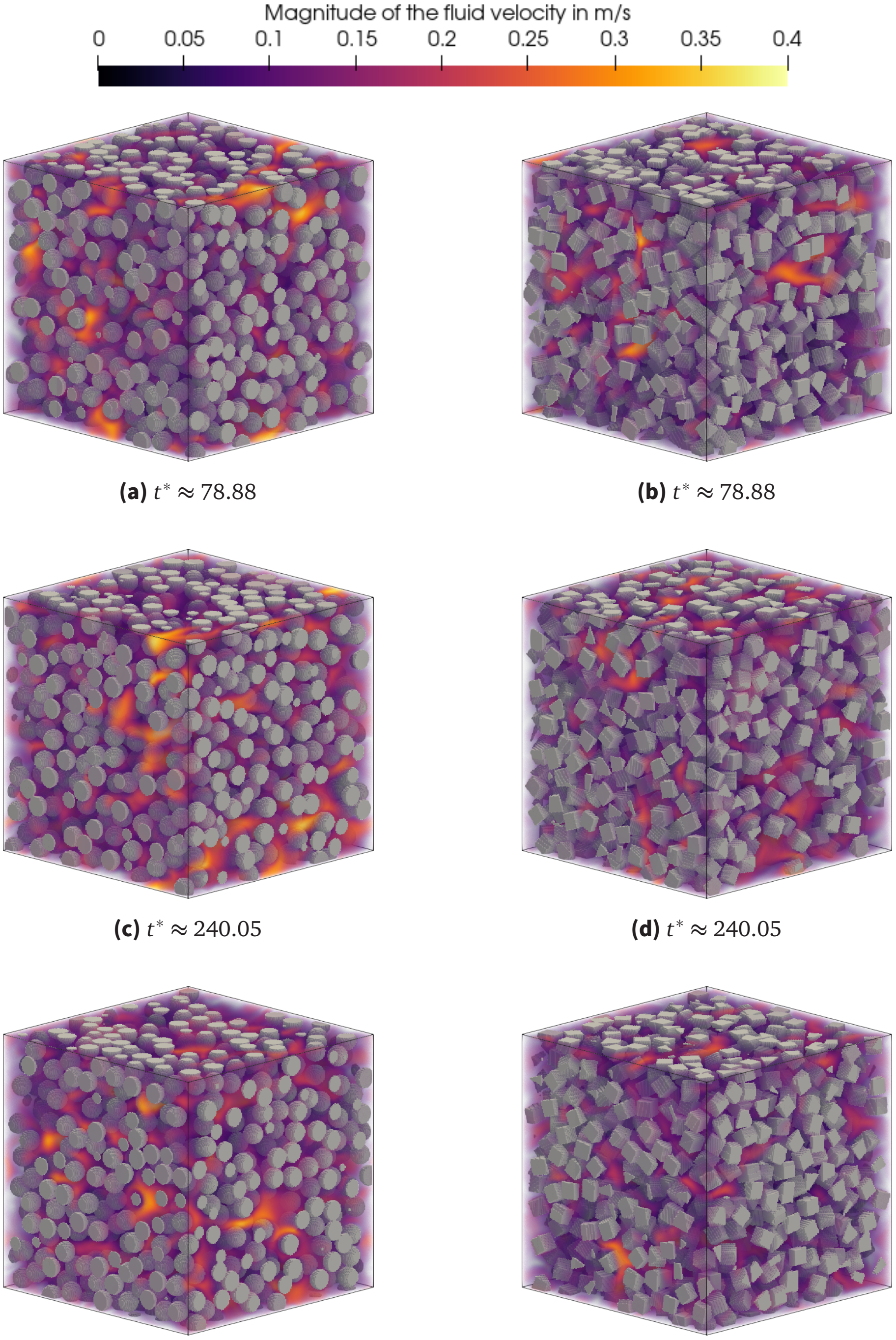


Figure 5.12: *Simulation domain with* 991 *cubic particles at their preassigned initial positions using a resolution of* $N = 27$ *cells per volume-equivalent sphere diameter and particle volume fraction* $\phi_p \approx 0.3$.

particles are also the same, as shown for a single parameter set in Figure 5.6 and Figure 5.12.

Figure 5.13 visualizes the fluid velocity field and particles for $Ar = 2000$ and $\phi_\mathrm{p} \approx 0.3$ at different times considering spheres (left) and cubes (right). In both cases, fluid velocities tend to be higher in regions of lower particle counts, i.e., in larger channels that form between clusters. However, we notice that the maximum velocities are higher when considering spheres, but the fluid velocity is more evenly distributed when considering cubes.

Table 5.1 compares the hindered settling behavior of a collective of spheres and cubes by displaying the average settling velocity of the spheres $\bar{u}_\mathrm{p,spheres}$ and cubes $\bar{u}_\mathrm{p,cubes}$ along with their absolute and relative differences and the corresponding particle volume fraction ϕ_p and Archimedes number Ar. For both geometries, the single sphere settling velocity is used for normalization u^*



(a) $t^* \approx 78.88$

(b) $t^* \approx 78.88$

(c) $t^* \approx 240.05$

(d) $t^* \approx 240.05$

(e) $t^* \approx 397.81$

(f) $t^* \approx 397.81$

Figure 5.13: *Velocity field around the particles at different normalized times t^* for $Ar = 2000$ and $\phi_p \approx 0.3$.*

Table 5.1: *Comparison of simulation results considering spheres and cubes at various Archimedes numbers Ar and particle volume fractions ϕ_p. The values in the table have been rounded. Discrepancies between calculated and rounded differences may occur due to rounding.*

Ar	ϕ_p	$\frac{\bar{u}_\mathrm{p,spheres}}{u^*}$	$\frac{\bar{u}_\mathrm{p,cubes}}{u^*}$	$\frac{\bar{u}_\mathrm{p,spheres}}{u^*} - \frac{\bar{u}_\mathrm{p,cubes}}{u^*}$	$\frac{\left(\bar{u}_\mathrm{p,spheres}/u^* - \bar{u}_\mathrm{p,cubes}/u^*\right)}{\left(\bar{u}_\mathrm{p,cubes}/u^*\right)}$
500	0.10	0.321	0.279	0.042	0.130
	0.15	0.277	0.226	0.050	0.182
	0.20	0.234	0.182	0.052	0.224
	0.25	0.196	0.147	0.049	0.250
	0.30	0.161	0.118	0.042	0.264
1000	0.10	0.274	0.236	0.038	0.140
	0.15	0.235	0.196	0.038	0.164
	0.20	0.202	0.161	0.041	0.202
	0.25	0.171	0.132	0.040	0.231
	0.30	0.144	0.106	0.038	0.264
2000	0.10	0.227	0.195	0.032	0.141
	0.15	0.193	0.164	0.029	0.151
	0.20	0.169	0.136	0.033	0.194
	0.25	0.145	0.113	0.032	0.220
	0.30	0.123	0.093	0.029	0.238

to ensure comparability.

Table 5.1 shows that the average settling velocity is higher for spherical particles in every case. Furthermore, the absolute difference seems to be approximately the same when the Archimedes number Ar remains constant. For $Ar = 500$, the absolute difference in the average settling velocities is the largest. As Ar increases, the absolute and relative difference becomes smaller. The relative difference decreases only slightly as Ar increases, but it is significantly smaller for low particle volume fractions.

The observed differences between the shapes are probably due to the fact that the cross-sectional area of the cubes changes with rotation, while the cross-sectional area of the spheres remains constant. The cubes have a higher drag coefficient, because they are rarely perfectly aligned. Also, the rotation of the cubes disturbs the fluid, which cannot flow around them as easily as it does around the rotationally invariant spheres. This seems to distribute the cubes more evenly, increasing the relative velocity between the fluid and the particles. In contrast to this, the spheres tend to form clusters more easily, occasionally allowing larger channels for the fluid to pass through.

The observation that the difference in the average settling velocity decreases with increasing *Ar*, which corresponds to increasing *Re*, supports the above assumption that cubes cause far more disturbance than spheres. As *Re* increases, so do the irregularity and chaos within the fluid. Therefore, the effect of more complex shapes becomes less important. We thus hypothesize that the geometry of the particle is irrelevant to turbulent flows.

The observation of a lower average settling velocity for cubes than for spheres is qualitatively consistent with literature. Similar findings have been made experimentally for cubes [20], rod-like particles [21], sand grains [22], and fibers [23]. This leads to the assumption that non-spherical particles generally experience greater hindered settling effects.

5.6 Summary and conclusions

In the present study, we introduce and verify a novel parallelization strategy tailored to the discrete contact model as outlined in the work of Marquardt et al. [71], along with a method to simultaneously handle periodic boundaries for four-way coupled surface resolved particles. For evaluation purposes, we use it together with HLBM, but it can also be used with any other PSM.

Our primary goal is to facilitate four-way coupled surface resolved particle simulations at high particle volume fractions using HLBM or other PSMs, while allowing arbitrary convex particle shapes. To confirm this, we perform extensive investigations of hindered settling of up to 1934 particles. Simulation predictions are compared quantitatively with established correlations for spherical particle collectives. In addition, we evaluate the computational cost of using this approach compared to simulations using the same HLBM implementation, but without contact handling. These results highlight the need for an explicit contact model, especially when dealing with high particle volume fractions, while maintaining efficient performance.

Furthermore, we perform preliminary hindered settling studies on cubes to demonstrate the applicability. The investigations carried out show a clear influence of the particle geometry on the bulk settling behavior. Swarms of cubes settle 13% slower at lower particle volume fractions and up to 26% slower at higher particle volume fractions compared to swarms of spheres. As *Ar* increases, however, the influence of the particle shape decreases slightly.

The aforementioned results emphasize the importance of the proposed model, since it is now possible to perform numerical experiments of particle flows at high particle volume fractions considering thousands of surface resolved arbitrarily shaped convex particles. This enables more accurate numerical simulations of real-world processes like filtration and thickening.

6
Engineering Applications and Future Prospects

This chapter has been published as

Marquardt, J. E. and Krause, M. J. "A Review of the Homogenized Lattice Boltzmann Method for Particulate Flow Simulations: From Fundamentals to Applications." In: Powders *3.4 (2024), pp. 500–530.* ISSN*: 2674-0516.* DOI*: 10.3390/powders3040027*

6.1 Introduction

The modeling and simulation of particulate flows are important in several academic and industrial areas within and beyond process engineering. For example, a better understanding of particulate flow dynamics contributes to advances in aerosol drug delivery [28], optimization of flow conditions for combustion processes to improve engine efficiency [132], and particle classification [133]. Process engineering research aims to understand the transformation processes of substances into other states, often described and modeled as particulate flows.

6.1.1 Overview of Particle-laden Flows Characteristics

Particle-laden or particulate flows are typically modeled as two-phase or two-component fluid flows characterized by the presence of two distinct dynamic phases or components. The first component, called the carrier fluid, forms a continuous phase, while the second component, called the dispersed phase or component, exists as discrete and separate entities [134]. Depending on the states of matter of the carrier and the dispersed components, several distinctions are made:

- Aerosols occur when the carrier component is a gas and the dispersed component is either a liquid (fog, mist) or a solid (dust, soot).
- Bubble flows occur when the carrier component is a liquid and the dispersed component is a dilute gas.
- Suspensions occur when the carrier component is a liquid and the dispersed component is a solid.
- Emulsions occur when both components are in a liquid state.

In foams, the gas phase is typically immobilized by a surrounding liquid or solid, resulting in a partly dynamic character. Therefore, foams are not classified as particulate flows.

If all particles within a dispersion have the same characteristic properties, the system is called monodisperse. Otherwise, it is referred to as polydisperse. These properties may be of physical, chemical, or organic origin. The Feret diameter is a common parameter that represents the maximum distance between two parallel tangential planes perpendicular to a given direction [135]. Occasionally, an average is calculated over several directions or a specific direction is chosen to give the minimum or maximum Feret diameter depending on the application.

Another size measure is the equivalent diameter, which is determined by comparing a particular property of an irregular particle to that of a regularly shaped particle. There are several types of equivalent diameters depending on the property chosen for comparison. The spherical equivalent diameter is particularly important in process engineering, where an irregularly shaped particle, such as a grain of sand, is related to a sphere of equivalent value by the diameter of the sphere. Other relevant properties in dynamic particle systems include the aspect ratio, volume, sphericity, or convexity [68]. Distributions of these shape parameters, along with other physical, chemical, or organic parameters, help to characterize a particle system and its dynamics.

Using a size parameter, such as the averaged Feret diameter, which considers the average size of the system across all or specific directions, allows for the differentiation between

- molecularly dispersed dissolved particles having a size of less than 1 nm,
- colloidally dispersed dissolved particles having a size between 1 nm and 1 µm,
- coarsely dispersed dissolved particles having a size greater than 1 µm.

Interactions among particles and between particles and the carrier component are commonly observed and must be considered. For colloidal and coarsely dispersed dissolved systems, the relevant scale ranges from 10^{-9} m to 10^{-4} m, which corresponds to the nano- and microscale. Since process engineering deals with transformation processes relevant to industrial applications [136], investigations are performed at process scales. Typical length scales in this domain range from 1 mm to 100 m. Consequently, sophisticated models are required to describe nano- or microscale interactions, such as agglomeration, separation, clustering, or phase changes, that affect process-scale suspension dynamics. The following example illustrates challenges and limitations.

In solid-liquid separation, effective separation of fine particle systems ranging from 100 nm to 10 µm remains a formidable challenge. Experiments at such scales are difficult to perform, and the lack of multiscale models and simulation tools makes experiments even more difficult. Despite efforts in various processes, such as centrifugal air classification [137] or cross-flow filtration [138], separation efficiencies remain insufficient. Within the size range considered, hydrodynamic, inertial, and particle interaction forces approach the equilibrium, rendering conventional methods ineffective.

Larger particles ($>$ 10 µm) allow for the use of inertial forces, as seen in centrifugal separators. Conversely, particles smaller than 100 nm show the dominance of diffusion and van-der-Waals forces. Here, techniques, such as gas-expanded liquids [139, 140], are suited for particle separation. Although some processes are applicable in this size range [141], their separation efficiency tends to be lower compared to the above methods due to the absence of predominant diffusion or inertia effects.

Chromatography [142] or ultra- and microfiltration [143, 144] are only suitable for minute quantities. The maximum throughput of cyclones is significantly limited by the particle diameter to be separated, e.g. for a particle diameter of 10 µm the maximum cyclone radius is 0.15 m [145].

Ensuring adequate separation efficiency, especially when dealing with oversized particles, is critical for many applications, such as wafer manufacturing. In chemical mechanical planarization, the presence of oversized particles in the polishing suspension renders the wafer unusable. The limitations inherent in the classification of fine particle systems underscore the critical need for a fundamental understanding of the interactions within these suspensions.

Complex models are essential for a detailed understanding of the dynamics of fine particle systems. These models serve as the basis for both physical and numerical experiments, providing insight into areas inaccessible by direct measurement. The study of particle-laden flows requires careful consideration of individual particle shapes, since they significantly influence the flow behavior [25, 146]. Moreover, practical applications typically involve polydisperse suspensions. Therefore, a comprehensive model capable of capturing different particle properties, in particular shape characteristics, as well as a simulation method and its implementation are crucial.

The following key aspects need to be addressed: First, flow simulations must strike a balance between detailing the influence of different particle shapes and ensuring computational efficiency. Second, realistic models and algorithms are needed to accurately describe particle dynamics. Third, thorough understanding, modeling, and accurate simulation of the coupling between the fluid and particle components are essential [147].

6.1.2 Overview of Modeling Approaches and Their Limitations

As discussed above, effective simulation of particulate flows requires a model that consists of at least three distinct submodels. The first submodel addresses the behavior of the carrier fluid, the second focuses on the dispersed component, and the third accounts for their interaction. Each subproblem is characterized by mathematical equations that are discretized and then translated into algorithms for computational implementation.

Within the length scales considered, i.e. > 100 nm, continuous Eulerian descriptions such as the Navier–Stokes, Stokes or Euler equations, are often used to describe the behavior of the carrier fluid.

At various scales, including nano-, micro-, and process scales, Lagrangian descriptions are commonly applied to characterize the dispersed component. This approach models discrete particles according to Newton's law of motion. At the atomistic scale, with diameters around 0.1 nm, the interaction between neutral pairs of atoms is typically described by the Lennard–Jones potential [148] or alternatively by the DLVO theory [149] which integrates van-der-Waals attraction, electrostatic repulsion, and Born repulsion phenomena. Beyond the atomic scale, molecular potentials become negligible. Instead, properties such as recoil forces due to deformation and frictional forces become important. These effects are often captured by the widely used spring-dashpot model [150].

In certain cases, the dispersed component can be effectively described by a continuous Euler formulation. In this approach, advection-diffusion equations are frequently adopted as the primary governing equations [28]. For monodisperse systems, a single equation proves adequate, whereas polydisperse systems require multiple equations to accurately represent the diverse particle sizes and distributions. This strategy is typically used to model diffusion or when the use of a Lagrangian approach becomes computationally prohibitive. The latter may occur when the number of suspended particles is too large to simulate individually or when the exact trajectory of each particle is not of primary interest. The computational cost associated with an Euler scheme scales primarily with the numerical resolution of the computational domain rather than with the number of particles. However, when considering nano- or microscale particles in process-scale applications, the number of particles to be simulated can easily exceed billions. In filtration, for example, scenarios assuming typical values for cake height and nutsche geometry can result in simulations involving approximately 30 billion particles. Despite their computational efficiency, Eulerian models often fail to capture finer details at the nano- or microscale because they rely on averaged quantities. They tend to neglect several potentially important properties, such as size or shape distributions, as well as inter-particle interactions.

If only the influence of the fluid on the particles is considered, the model is called one-way coupled. A model is classified as two-way coupled, if it assumes mutual influence between the fluid and the particles, and vice versa. It thus introduces back-coupling. Four-way coupled models consider interactions between the particles themselves as well as particle-wall interactions, providing a more comprehensive representation of the dynamic interactions within the system. The coupling approaches vary depending on the model used for the dispersed component.

In Lagrangian models, hydrodynamic forces are typically computed as integral forces for fully resolved particles or they are modeled using drag laws, such as Stokes, Kaskan, Kürten, or Martin, for unresolved particles of simpler shapes, including spheres and ellipsoids [151]. These models often assume a dilute regime, implying the absence of other particles. Sophisticated models for denser situations rely on heuristics and measurements and are usually only applicable to simpler scenarios, such as swarm sedimentation of spheres [77]. These latter approaches are known as subgrid-scale methods, where particles are described as points rather than volumetric domains, while transporting properties, such as mass, momentum, energy, and others.

When two-way coupling is required, adjustments to the fluid dynamics equations are essential. In fully resolved simulations, the fluid domain changes dynamically with time due to particle motion. For subgrid-scale models, modeling back-coupling typically involves extending the Navier–Stokes equations to the volume-averaged Navier–Stokes equations (VANSE) [152], incorporating a dynamic solid volume fraction. The challenges in both back-coupling approaches arise primarily in the numerical implementation rather than in the modeling phase.

The coupling of two Eulerian models follows an approach similar to that of subgrid-scale Lagrangian models. In addition, a drag model is required that imposes similar restrictions on simpler shaped particles in a dilute regime as described above. Back-coupling can also be achieved using VANSE, but requires the provision of diffusion and convection coefficients. Currently, closure approaches still rely on measurements or heuristics and models are accessible only for very simple suspensions. A representation for complex particle shapes may be lacking [147].

6.1.3 Overview of Numerical Solution Approaches and Their Limitations

A widely used numerical approach for Lagrangian models, especially for simulations involving large numbers of particles (> 1000), is the discrete element method (DEM). In its conventional form, DEM treats particles as ideal spheres. A comprehensive review of DEM can be found in the works of Zhu et al. [32, 33]. This subgrid-scale approach treats particles as mass points whose dynamics are governed by applied forces, such as drag, according to Newton's second law. These methods typically use explicit time discretization schemes, such as forward Euler, which require small time steps. Consequently, the simulation involves a separation of free flight and collisions, where the forces acting on the particles adjust their velocity for the next time step. Extensions of the DEM have been developed to accommodate ellipsoidal [34–37] and polygonal [38–40] particle shapes. However, as the complexity of the particle shapes increases, so does the computational complexity of the simulations.

More sophisticated DEM approaches combine convex shapes. For example, the multi-sphere approach approximates arbitrary particle shapes by assembling multiple spheres [41, 42]. Other convex shapes can also be used [44]. In such approaches, forces acting on each element associated with the particle are computed and aggregated to characterize the behavior of the entire object. However, accurately representing arbitrarily shaped particles requires a significant number of spheres or complex subelements, resulting in significant computational requirements.

In recent years, the Immersed Boundary Method (IBM) has received considerable attention as a numerical technique for integrating fully resolved Lagrangian particles with Eulerian fluid descriptions. This method uses a discrete set of Lagrangian points to represent the surface of particles within an Eulerian mesh [57, 58], allowing the simulation of particles with arbitrary shapes. However, this approach requires a sufficiently fine resolution of the computational grid to accurately capture interactions between the Eulerian mesh and the Lagrangian points by interpolation. The combination with modern collision detection algorithms [37, 60] leads to high computational costs.

In Lagrangian methods, the treatment of particle contact is a major challenge. While computationally efficient spring-dashpot models, i.e. the Hertz-Mindlin model, are typically applicable to spherical particles only [150], handling more complex shapes leads to higher computational costs. Common models are limited to polygonal [34, 110, 153] or convex [60] particles. Only recent developments allow for the modeling of more complex shapes [44, 71, 105, 123], but their application to large-scale problems with thousands of particles and four-way coupling remains challenging.

In the case of IBM, back-coupling is inherently integrated, while in the case of DEM, it is not. Zhang et al. [154] introduced an IBM-based approach that relies on pulse exchange using the lattice Boltzmann method (LBM). However, this method requires further calibration through additional simulations and has been shown to lack convergence for single sphere sedimentation problems [155]. Another LBM-based approach proposed by Blais et al. [152] solves the VANSE, but it also exhibits convergence problems in dynamic scenarios, although these could potentially be solved using recently proposed methods [156, 157].

Another numerical approach to simulating particle-laden flows uses Eulerian models in which the particles are represented as distributions. Discretization methods based on LBM include the multi-phase/component methods developed by Shan and Chen [158] and methods that discretize the advection-diffusion equation [27, 159]. The latter uses a one-way coupling approach and incorporates a Stokes drag law. This method has been extended to a two-way coupling approach by introducing an LBM-based discretization for the VANSE [156, 157]. While these models offer efficient computation, they are limited by their relatively low level of detail. Currently, these models allow for the inclusion of population balances and for the capture of particle size distributions, inertial forces, and particle interactions [160, 161]. However, Eulerian models for determining particle shape properties or their distribution are not yet available.

In conclusion, there is currently a lack of methods capable of simulating large numbers of arbitrarily shaped particles. Some approaches lack the required level of detail regarding shape effects, while others struggle with their computational demands. Therefore, there is a need to develop, test, and implement new methods that meet these criteria [147].

6.1.4 Aims of the Homogenized Lattice Boltzmann Method

The aim of this paper is to comprehensively review the homogenized lattice Boltzmann method (HLBM) [66, 67], which was introduced specifically to address the above challenges. HLBM's strengths lie in its monolithic nature, which seamlessly accommodates pure fluid dynamics, porous media, and moving solid and porous particles. It excels at representing arbitrary particle shapes, provides surface-resolved simulations including detailed surface force calculations, and operates efficiently in parallel, thus accelerating computations. By providing a detailed investigation of HLBM and its capabilities of simulating complex particle-laden flows, this review aims to contribute to the advancement of numerical methods in this field.

6.1.5 Structure of This Review Paper

Section 6.2 provides details on the underlying models, while Section 6.3 explains the numerical methods used. Section 6.4 presents reported applications of HLBM, so that the strengths and weaknesses as well as future development needs can be discussed in Section 6.5. Finally, Section 6.6 summarizes and concludes the review.

6.2 Modeling

This section describes the numerical study of particulate flows by applying a fully resolved Eulerian-Lagrangian approach to the problem. The essential parts are the fluid (denoted by f), a particle set (denoted by P) consisting of individual particles p ∈ P, and the interaction between these components. The approach is introduced in the following sections, where we consider a fluid domain $\Omega_{\mathrm{f}}(t)$ and particle domains $\Omega_{\mathrm{P}}(t) = \bigcup_{\mathrm{p}\in\mathrm{P}} \Omega_{\mathrm{p}}(t)$. The entire computational domain is thus given by $\Omega = \Omega_{\mathrm{f}}(t) \cup \Omega_{\mathrm{P}}(t)$. Note that both the fluid and particle domains are time-dependent, while Ω remains constant.

6.2.1 Fluid Dynamics

The fluid dynamics model is based on the conservation laws of classical hydrodynamics. Firstly, the principle of mass conservation reads [162]

$$\frac{\partial \rho_{\mathrm{f}}}{\partial t} + \nabla \cdot (\rho_{\mathrm{f}} \boldsymbol{u}_{\mathrm{f}}) = 0, \tag{6.1}$$

with the fluid density ρ_{f}, velocity $\boldsymbol{u}_{\mathrm{f}}$, and dynamic viscosity η. Secondly, the conservation of momentum is described using Cauchy's equation of motion [163]

$$\frac{\partial \rho_{\mathrm{f}} \boldsymbol{u}_{\mathrm{f}}}{\partial t} + \rho_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{f}} \cdot \nabla)\boldsymbol{u}_{\mathrm{f}} = \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{F}_{\mathrm{f}}, \tag{6.2}$$

where $\boldsymbol{\sigma}$ denotes the Cauchy stress tensor [163] representing surface forces and $\boldsymbol{F}_{\mathrm{f}}$ is the total of all volume forces acting on the fluid.

In case of a Newtonian fluid with uniform density ρ_{f} and temperature-independent volume forces $\boldsymbol{F}_{\mathrm{f}}$, the above conservation laws simplify to the incompressible Navier–Stokes equations [162]

$$\begin{aligned} \nabla \cdot \boldsymbol{u}_{\mathrm{f}} &= 0, \\ \rho_{\mathrm{f}}\left(\frac{\partial \boldsymbol{u}_{\mathrm{f}}}{\partial t} + (\boldsymbol{u}_{\mathrm{f}} \cdot \nabla)\boldsymbol{u}_{\mathrm{f}}\right) &= -\nabla p + \eta \Delta \boldsymbol{u}_{\mathrm{f}} + \boldsymbol{F}_{\mathrm{f}}, \end{aligned} \tag{6.3}$$

with the pressure p.

Coupling from a solid component to the fluid is introduced by adding a linear body force term according to Spaid and Phelan [164]

$$\boldsymbol{F}(\boldsymbol{x}, t) = -\eta \boldsymbol{K}^{-1}(\boldsymbol{x}, t)\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x}, t), \tag{6.4}$$

which introduces the influence of a porous medium with the permeability $\boldsymbol{K}$. By assuming isotropy and periodicity, $\boldsymbol{K}$ reduces to its only eigenvalue K. This leads to the homogenized Navier–Stokes equations [147, 165], in which the momentum equation becomes

$$\rho_{\mathrm{f}}\left(\frac{\partial \boldsymbol{u}_{\mathrm{f}}}{\partial t}+(\boldsymbol{u}_{\mathrm{f}}\cdot\nabla)\boldsymbol{u}_{\mathrm{f}}\right)=-\nabla p+\eta\Delta\boldsymbol{u}_{\mathrm{f}}-\eta K^{-1}\boldsymbol{u}_{\mathrm{f}}+\boldsymbol{F}_{\mathrm{f}}. \tag{6.5}$$

The permeability-dependent term in the applied body force follows

$$K^{-1}(\boldsymbol{x},t)=\begin{cases}K_{\mathrm{p}}^{-1}, & \text{if } \boldsymbol{x}\in\Omega_{\mathrm{p}}(t)\\ 0, & \text{otherwise}\end{cases}, \tag{6.6}$$

which means that the force disappears outside of particles or depends on the permeability K_{p} of the particle intersecting the position.

Note that Guo and Zhao [166] criticize the lack of a nonlinear inertial and convective term. Therefore, they include a Forchheimer term leading to

$$\boldsymbol{F}(\boldsymbol{x},t)=-\frac{\eta}{K}\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x},t)-\frac{\epsilon}{\sqrt{K}}\frac{1.75}{\sqrt{150\epsilon^{3}}}||\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x},t)||\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x},t), \tag{6.7}$$

with the porosity ϵ. However, in previous studies of particulate flows, the model of Spaid and Phelan [164] has led to accurate results as shown in Section 6.4.

6.2.2 Particle Dynamics

For the particle component, Newton's second law of motion is used to describe the particles' translation

$$m_{\mathrm{p}}\frac{\partial \boldsymbol{u}_{\mathrm{p}}}{\partial t}=\boldsymbol{F}_{\mathrm{p}}, \tag{6.8}$$

and rotation

$$\boldsymbol{I}_{\mathrm{p}}\frac{\partial \boldsymbol{\omega}_{\mathrm{p}}}{\partial t}+\boldsymbol{\omega}_{\mathrm{p}}\times(\boldsymbol{I}_{\mathrm{p}}\cdot\boldsymbol{\omega}_{\mathrm{p}})=\boldsymbol{T}_{\mathrm{p}}. \tag{6.9}$$

The symbols m_{p}, $\boldsymbol{I}_{\mathrm{p}}$, $\boldsymbol{u}_{\mathrm{p}}$, and $\boldsymbol{\omega}_{\mathrm{p}}$ represent the particle's mass, moment of inertia, velocity, and angular velocity, respectively. $\boldsymbol{F}p$ and $\boldsymbol{T}_{\mathrm{p}}$ denote the total force and torque acting on the particle, respectively. They refer to the particle's center of mass $\boldsymbol{X}_{\mathrm{p}}$.

The forces acting on the particle can also be divided into surface forces and volume forces. The latter are caused by gravity and buoyancy, which are commonly included via their net force $\boldsymbol{F}_{\mathrm{p,V}} = (\rho_{\mathrm{f}} - \rho_{\mathrm{p}})\, V_{\mathrm{p}}\boldsymbol{g}$, with the particle density ρ_{p}, the particle volume V_{p}, and the gravitational acceleration vector $\boldsymbol{g}$.

The surface forces consist of contact forces $\boldsymbol{F}_{\mathrm{p,c}}$ and hydrodynamic forces $\boldsymbol{F}_{\mathrm{p,h}}$, meaning

$$\boldsymbol{F}_{\mathrm{p}} = \boldsymbol{F}_{\mathrm{p,V}} + \boldsymbol{F}_{\mathrm{p,c}} + \boldsymbol{F}_{\mathrm{p,h}} = \boldsymbol{F}_{\mathrm{p,V}} + \boldsymbol{F}_{\mathrm{p,c}} + \int\limits_{\Gamma_{\mathrm{p}}} \boldsymbol{\sigma} \cdot \boldsymbol{n}_{\mathrm{p},\Gamma}\, d\Gamma_{\mathrm{p}}, \tag{6.10}$$

where the hydrodynamic forces are related to the above stress tensor $\boldsymbol{\sigma}$ [70, 162]. The particle's surface is denoted by Γ_{p} and surface normal vectors by $\boldsymbol{n}_{\mathrm{p},\Gamma}$.

Similarly, the total torque reads

$$\boldsymbol{T}_{\mathrm{p}} = \boldsymbol{T}_{\mathrm{p,c}} + \boldsymbol{T}_{\mathrm{p,h}} = \sum_{p \neq q} (\boldsymbol{x}_{\mathrm{c,pq}} - \boldsymbol{X}_{\mathrm{p}}) \times \boldsymbol{F}_{\mathrm{c,pq}} + \int\limits_{\Gamma_{\mathrm{p}}} \left((\boldsymbol{x}_{\Gamma} - \boldsymbol{X}_{\mathrm{p}}) \times \boldsymbol{\sigma} \right) \cdot \boldsymbol{n}_{\mathrm{p},\Gamma}\, d\Gamma_{\mathrm{p}}, \tag{6.11}$$

which uses the above relationship between the hydrodynamic force and the stress tensor $\boldsymbol{\sigma}$, the center of mass of the particle $\boldsymbol{X}_{\mathrm{p}}$, and the individual contact forces $\boldsymbol{F}_{\mathrm{c,pq}}$ with their corresponding contact points $\boldsymbol{x}_{\mathrm{c,pq}}$ as well as the surface points $\boldsymbol{x}_{\Gamma}$. See Section 6.2.3 below for details on the formulation of the contact forces.

6.2.3 Contact

A common approach to account for solid-solid interactions is to use overlaps to model deformations that occur in reality, which is only valid for very small overlaps. The total contact force

$$\boldsymbol{F}_{\mathrm{p,c}} = \sum_{p \neq q} \boldsymbol{F}_{\mathrm{c,pq}} \tag{6.12}$$

acting on a particle p is the sum of all separate forces caused by individual contacts between that particle and any other solid q. Here, the subscript q represents each distinct solid object other than particle p. The individual contact forces usually consist of a normal and a tangential component, i.e.

$$\boldsymbol{F}_{\mathrm{c,pq}} = \boldsymbol{F}_{\mathrm{c,pq,n}} + \boldsymbol{F}_{\mathrm{c,pq,t}}. \tag{6.13}$$

Nassauer's and Kuna's model [105] provides equations for both normal

$$\boldsymbol{F}_{\mathrm{c,pq,n}} = k E^{*}_{\mathrm{pq}} \sqrt{V_{\mathrm{pq}} d_{\mathrm{pq}}} \left(1 + c_{\mathrm{pq}} \dot{d}_{\mathrm{pq,n}} \right) \boldsymbol{n}_{\mathrm{c,pq}}, \tag{6.14}$$

and tangential contact forces

$$\boldsymbol{F}_{\mathrm{c,pq,t}} = -\frac{\boldsymbol{u}_{\mathrm{pq,t}}(\boldsymbol{x}_{\mathrm{c,pq}})}{||\boldsymbol{u}_{\mathrm{pq,t}}(\boldsymbol{x}_{\mathrm{c,pq}})||}\left(\left(2\mu^*_{\mathrm{pq,s}} - \mu_{\mathrm{pq,k}}\right)\frac{a^2_{\mathrm{pq}}}{a^4_{\mathrm{pq}}+1} + \mu_{\mathrm{pq,k}} - \frac{\mu_{\mathrm{pq,k}}}{a^2_{\mathrm{pq}}+1}\right)||\boldsymbol{F}_{\mathrm{c,pq,n}}||, \tag{6.15}$$

with

$$\mu^*_{\mathrm{pq,s}} = \mu_{\mathrm{pq,s}}\left(1 - 0.09\left(\frac{\mu_{\mathrm{pq,k}}}{\mu_{\mathrm{pq,s}}}\right)^4\right), \tag{6.16}$$

and

$$a_{\mathrm{pq}} = \frac{||\boldsymbol{u}_{\mathrm{pq,t}}(\boldsymbol{x}_{\mathrm{c,pq}})||}{u_k}. \tag{6.17}$$

The normal force considers factors like the effective Young's modulus

$$E^*_{\mathrm{pq}} = \left(\frac{1-\nu^2_{\mathrm{p}}}{E_{\mathrm{p}}} + \frac{1-\nu^2_{\mathrm{q}}}{E_{\mathrm{q}}}\right)^{-1}, \tag{6.18}$$

a constant k, overlap volume V_{pq}, indentation depth d_{pq}, damping factor c_{pq}, and relative velocity $\dot{d}_{\mathrm{pq,n}}$ along the contact normal $\boldsymbol{n}_{\mathrm{c,pq}}$.

In sphere–half-space and sphere–sphere contacts, the constant k is set to $4/(3\sqrt{\pi})$ [105]. Conversely, for a cylindrical flat punch, k has the value $2/\sqrt{\pi}$ [71]. The effective Young's modulus is calculated from the Young's modulus E and Poisson's ratio ν of the objects p and q, while the damping factor c_{pq} correlates with the signed coefficient of restitution e_{pq} based on Carvalho's and Martins' relationship [106]

$$c_{\mathrm{pq}} = \begin{cases} 1.5\frac{(1-|e_{\mathrm{pq}}|)(11-|e_{\mathrm{pq}}|)}{(1+9|e_{\mathrm{pq}}|)u_0} & \text{if } u_0 > 0 \\ 0 & \text{if } u_0 = 0 \end{cases}, \tag{6.19}$$

using the magnitude of the initial relative velocity $u_0 \geq 0$ [88].

Tangential forces influenced by static $\mu_{\mathrm{pq,s}}$ and kinetic $\mu_{\mathrm{pq,k}}$ friction coefficients are described by Eq. (6.15), involving the relative tangential velocity between the objects p and q denoted by $\boldsymbol{u}_{\mathrm{pq,t}}$ at the contact point $\boldsymbol{x}_{\mathrm{c,pq}}$, while the model parameter u_k quantifies the velocity at which the shift from static to kinetic friction takes place.

6.3 Numerical Methods

In the next section, the numerical methods used to solve the above models are presented. Note that all values discussed in Sections 6.3.1 and 6.3.2 are expressed in lattice units.

6.3.1 Lattice Boltzmann Method

The lattice Boltzmann method (LBM) [90, 91] is a well-established and robust option to solve the incompressible Navier–Stokes equations. LBM operates on a mesoscopic level, discretizing both spatial and temporal domains by employing fluid particle distributions to characterize fluid flow dynamics. These distributions, symbolized as $f_i(\boldsymbol{x},t)$, depict the likelihood of encountering particles possessing discrete velocities $\boldsymbol{c}_i$ at a given position $\boldsymbol{x}$ and time t. Various velocity sets have been proposed and extensively discussed in existing literature [90]. Studies using HLBM typically use the D2Q9 velocity set for two-dimensional setups [62, 66], which contains 9 discrete velocities in a two-dimensional space, while for a three-dimensional setup [67, 69], D3Q19 is used, where the space is three-dimensional and the velocity set contains 19 velocities.

LBM computationally determines fluid flow by iteratively updating fluid particle distributions using two steps. Firstly, the collision step manages local interparticle interactions. It is expressed as

$$f_i^*(\boldsymbol{x},t) = f_i(\boldsymbol{x},t) + Q_i(\boldsymbol{x},t) + S_i(\boldsymbol{x},t). \tag{6.20}$$

Here, f_i^* denotes the post-collision distribution, Q_i the collision operator, and S_i an optional source term. Secondly, the propagation step distributes the fluid particles to neighboring lattice nodes and is given by

$$f_i(\boldsymbol{x} + \Delta t \boldsymbol{c}_i, t + \Delta t) = f_i^*(\boldsymbol{x},t), \tag{6.21}$$

where $\Delta t = \Delta x = 1$ represent the time step size and grid spacing, respectively.

A variety of collision operators exist [90]. In combination with HLBM, however, the Bhatnagar–Gross–Krook (BGK) collision operator [92] is used exclusively, which reads

$$Q_i(\boldsymbol{x},t) = -\frac{1}{\tau}(f_i(\boldsymbol{x},t) - f_i^{\mathrm{eq}}(\rho_{\mathrm{f}}, \boldsymbol{u}_{\mathrm{f}})), \tag{6.22}$$

where τ is the relaxation time that governs the relaxation towards an equilibrium distribution f_i^{eq}. The state of equilibrium is characterized by the Maxwell–Boltzmann distribution and expressed as [93]

$$f_i^{\mathrm{eq}}(\rho_{\mathrm{f}}, \boldsymbol{u}_{\mathrm{f}}) = w_i \rho_{\mathrm{f}} \left(1 + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}_{\mathrm{f}}}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}_{\mathrm{f}})^2}{2c_s^4} - \frac{\boldsymbol{u}_{\mathrm{f}} \cdot \boldsymbol{u}_{\mathrm{f}}}{2c_s^2}\right). \tag{6.23}$$

The associated weights w_i are obtained from a Gauss-Hermite quadrature rule and remain constant for the chosen velocity set. Meanwhile, the constant lattice speed of sound c_s also relies on the specific velocity set selected for the simulation and is additionally used to determine τ from the fluid's viscosity via $\tau = (1/c_s^2)(\eta/\rho_\mathrm{f}) + 0.5$.

Utilizing the above fluid particle distributions allows for the derivation of macroscopic quantities [90], including the fluid density $\rho_\mathrm{f}(\boldsymbol{x}, t) = \sum_i f_i(\boldsymbol{x}, t)$, which is related to the pressure via $p = \rho_\mathrm{f} c_s^2$ and momentum $\rho_\mathrm{f}\boldsymbol{u}_\mathrm{f}(\boldsymbol{x}, t) = \sum_i \mathbf{c}_i f_i(\boldsymbol{x}, t)$.

6.3.2 Homogenized Lattice Boltzmann Method

To account for the moving porous media, i.e. to solve the homogenized Navier–Stokes equations, see Eq. (6.5), HLBM is used. It couples porous particles to the fluid by introducing a continuous model parameter $B(\boldsymbol{x}, t) \in [0, 1]$, which is mapped onto the whole computational domain using a trigonometric level set function [71]

$$B(\boldsymbol{x}, t) = \begin{cases} 0, & \text{if } d_s \leq -\varepsilon/2 \\ \cos^2\left(\frac{\pi}{2}\left(\frac{d_s}{\varepsilon} - \frac{1}{2}\right)\right), & \text{if } \varepsilon/2 > d_s > -\varepsilon/2 \,. \\ 1, & \text{if } d_s \geq \varepsilon/2 \end{cases} \tag{6.24}$$

Here, ε is a smoothing parameter, while d_s represents the signed distance to this boundary surface. Figure 6.1 illustrates this concept using a circular boundary, showing the smooth transition region with its width defined by ε, which is of the scale of the lattice spacing Δx. For $\varepsilon \to 0$, the transition region would vanish, resulting in a sharp boundary [66].

Note that STL files are also usable for particle shape description, for example from the online particle database PARROT [167]. For such files, the signed distance is then cached in a second mesh with a predefined spacing smaller than or equal to Δx, which is then used to interpolate the dynamic signed distances [67, 168].

In the context of HLBM, this level set function is also called confined permeability and is related to the permeability of the medium K by

$$B(\boldsymbol{x}, t) = 1 - \tau \frac{\eta}{\rho_\mathrm{f} K}. \tag{6.25}$$

Here, τ is the relaxation time, which also appears in the collision operator in Eq. (6.22). This leads to a dependence on the numerical setup and constrains the resolvable permeability K to the range $[\tau\eta/\rho_\mathrm{f}, \infty)$ [86].

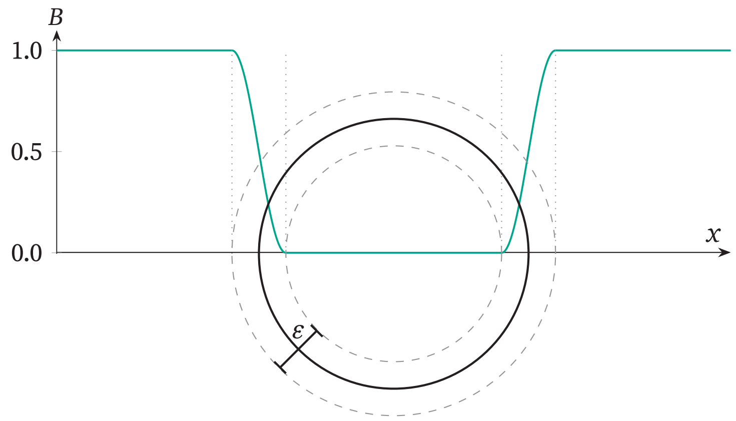


Figure 6.1: *Representation of the result of Eq. (6.24) along an x-axis through the middle of a circle (black solid line), illustrating the smooth transition layer with size ε as dashed circles.*

The confined permeability is used to couple the particle to the fluid, by the convex combination of velocities

$$\Delta \boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x},t) = (B(\boldsymbol{x},t)-1)\big(\boldsymbol{u}_{\mathrm{f}}(\boldsymbol{x},t)-\boldsymbol{u}_{\mathrm{p}}(\boldsymbol{x},t)\big) \tag{6.26}$$

in an adaptation of the exact difference method (EDM) by Kupershtokh et al. [95], which introduces the source term

$$S_i(\boldsymbol{x},t) = f_i^{\mathrm{eq}}(\rho_{\mathrm{f}},\boldsymbol{u}_{\mathrm{f}}+\Delta\boldsymbol{u}_{\mathrm{f}}) - f_i^{\mathrm{eq}}(\rho_{\mathrm{f}},\boldsymbol{u}_{\mathrm{f}}) \tag{6.27}$$

in Eq. (6.20) [69].

In previous publications, coupling of the particle was also considered part of HLBM, for which the momentum exchange algorithm (MEA) of Wen et al. [96] has shown to provide the most accurate results [69]. Using the MEA, the hydrodynamic force is calculated per lattice node using

$$\boldsymbol{F}_{\mathrm{p,h}}^{\mathrm{local}}(\boldsymbol{x},t) = \sum_i (\boldsymbol{c}_i - \boldsymbol{u}_{\mathrm{p}}(\boldsymbol{x},t)) f_i(\boldsymbol{x}+\Delta t \boldsymbol{c}_{\bar{i}},t) - (\boldsymbol{c}_{\bar{i}} - \boldsymbol{u}_{\mathrm{p}}(\boldsymbol{x},t)) f_{\bar{i}}(\boldsymbol{x},t). \tag{6.28}$$

$\bar{i}$ denotes particle distributions characterized by the discrete velocity $\boldsymbol{c}_{\bar{i}} = -\boldsymbol{c}_i$, indicating that the distribution $f_{\bar{i}}$ has an opposite orientation to f_i. The sum of all forces on nodes inside the particle, which are identified by the position $\boldsymbol{x}_{\mathrm{b}}$, leads to the total hydrodynamic force acting on the particle

$$\boldsymbol{F}_{\mathrm{p,h}}(t) = \sum_{\boldsymbol{x}_{\mathrm{b}}} \boldsymbol{F}_{\mathrm{p,h}}^{\mathrm{local}}(\boldsymbol{x}_{\mathrm{b}},t). \tag{6.29}$$

This results in the torque

$$\boldsymbol{T}_{\mathrm{p,h}}(t) = \sum_{\boldsymbol{x}_{\mathrm{b}}} (\boldsymbol{x}_{\mathrm{b}} - \boldsymbol{X}_{\mathrm{p}}) \times \boldsymbol{F}^{\mathrm{local}}_{\mathrm{p,h}}(\boldsymbol{x}_{\mathrm{b}}, t). \tag{6.30}$$

Using the above equations for hydrodynamic force and torque, along with the contact forces, see Sections 6.2.3 and 6.3.3, gravity, and buoyancy, the total force and torque are calculated with the help of Eqs. (6.10) and (6.11). The forces are converted into accelerations according to Newton's second law of motion, which are then used in the velocity Verlet algorithm [108] to solve Eqs. (6.8) and (6.9) and to obtain the particle velocity $\boldsymbol{u}_{\mathrm{p}}$ and then the position of the particle's center of mass $\boldsymbol{X}_{\mathrm{p}}$.

6.3.3 Discrete Contacts

The discrete contact model [71] consists of three fundamental mesh-based procedures: Rough contact detection, subsequent correction, and the computation of resultant forces.

While mapping $B(\boldsymbol{x}, t)$ onto the domain, initial rough contact detection is performed. For this, the signed distance of particles is evaluated at each lattice node. If the signed distance between two particles at a node is less than half the diagonal of a cell, i.e. $d_{\mathrm{s}} < \sqrt{0.5}\Delta x$ in two dimensions and $d_{\mathrm{s}} < \sqrt{0.75}\Delta x$ in three dimensions, a potential contact is detected. This preliminary estimate will produce an approximate cuboid that encapsulates the contact area, forming a bounding box.

Due to the relatively large lattice spacing, however, the overlap region is insufficiently resolved, resulting in an imprecise bounding box. To overcome this limitation, a subsequent correction step is implemented to refine the bounding box and align it more accurately with the actual overlap. This correction involves iteratively traversing the surface of the initial approximation using a predefined number of points along each spatial direction, known as the contact resolution N_{c}. At each point, discrete directional computations are performed to measure the distance to the true contact, thereby increasing the accuracy of the bounding box.

In the final step, the contact resolution N_{c} is used again for systematic iteration over the entire overlap region to determine the overlap volume, contact point, contact normal, indentation depth, and contact force.

For a full understanding of the details of the algorithm, readers are encouraged to consult the relevant publications [71, 73].

6.4 Applications

Several publications extended HLBM or used it for studies in the past. A chronological overview is given in Table 6.1, starting with the oldest publication. As can be seen, these publications generally follow a very logical order, starting with relatively simple cases, i.e. 2D simulations [66], and then continuously adding complexity.

The initial proposal of the method [66] was followed by allowing the consideration of arbitrarily shaped three-dimensional particles for two-way coupled simulations [67] and later by adding a compatible contact model for convex particles [71] in order to also consider higher particle volume fractions, as previous studies showed the need for an explicit contact model in these cases [69].

After the introduction of the necessary models, together with extensive validations, HLBM has been used to study industrially relevant processes. For example, studies covered a shape-dependent drag correlation [68, 170], the application to wall-flow filters [86–88, 129], hindered settling [72, 73], as well as the damage potential of particles during transport through cross-sectional

Table 6.1: *Overview of publications on HLBM.*

Publication	Spatial Dimensions	Particle Shapes	Particle Number per Simulation	Coupling
Krause et al. [66]	2	Circles, squares, and triangles	1 – 24	Two- and four-way
Trunk et al. [67]	3	Spheres and limestone CT scans	1 – 15	Two-way
Dapelo et al. [169]	3	Spheres	1	Two-way
Haussmann et al. [62]	2	Circles	1	Two-way
Bretl et al. [170]	3	Symmetric regarding the principal planes	1	Two-way
Trunk et al. [69]	2 and 3	Circles and spheres	1 – 1865	Two-way
Trunk et al. [68]	3	Superellipsoids	1	Two-way
Hafen et al. [86]	3	Cubic disks	48 – 240	Two-way
Marquardt et al. [71]	3	Spheres	1	Four-way
Hafen et al. [129]	3	Cubic disks	1	Two-way
Hafen et al. [87]	3	Cubic disks	1 – 480	Two-way
Hafen et al. [88]	3	Cubic disks	240 – 720	Four-way
Marquardt et al. [72]	3	Spheres	166 – 991	Two-way

Table 6.1: Cont.

Publication	Spatial Dimensions	Particle Shapes	Particle Number per Simulation	Coupling
Marquardt et al. [73]	3	Spheres and cubes	191 – 1934	Four-way
Marquardt et al. [171]	3	Cubes	394 – 784	Four-way

constrictions in pipes [171].

As research using HLBM is extensive, we focus on the most recent and relevant studies using HLBM and their results to provide an overview of the applicability of HLBM below.

This includes the flow around a cylinder in Section 6.4.1, the settling of a single sphere in Section 6.4.2, and shape-dependent single particle settling in Section 6.4.3. Furthermore, we demonstrate the incorporation of the above discrete contact model by showing the results of a cylinder-wall impact in Section 6.4.4 and a spherical particle rebound in Section 6.4.5. Finally, we consider the results of the complex wall-flow filter in Section 6.4.6, the investigation of the hindered settling phenomena in Section 6.4.7, and mechanical stress during transport through cross-section constrictions in Section 6.4.8.

All of the above publications and the applications shown below used the open source software OpenLB [94, 168].

6.4.1 Flow around a Cylinder

To ensure convergence of the hydrodynamic forces $\boldsymbol{F}_\mathrm{h}$, Trunk et al. [67] considered a flow around a cylinder following Schäfer et al. [172]. The corresponding simulation setup is shown as a projection onto the x-y plane in Figure 6.2. The cylinder fixed at the beginning of the channel is modeled with HLBM and has a length of 0.41 m, which is equal to the channel length in the z-direction. The lattice spacing is $\Delta x = 0.01\mathrm{m}/N$ and the fluid flows from the left to the right.

The walls were modeled with a bounce back boundary. The inlet on the left ($x = 0$ m) was set as a regularized velocity boundary, for which the velocity profile is given by [172]

$$\boldsymbol{u}_\mathrm{f}(y,z) = 7.2yz\frac{(0.41-y)(0.41-z)}{0.41^4}. \tag{6.31}$$

At the outlet on the right, a regularized pressure boundary was used. Details of the boundary conditions can be found in literature [173]. The considered fluid had the viscosity $\eta = 1$ Pa s and density $\rho_\mathrm{f} = 1000$ kg/m^3. Using the average inflow velocity $\bar{u}_{\mathrm{f},0}$, the drag coefficient is defined as

$$C_\mathrm{d} = \frac{2F_{\mathrm{h},x}}{0.041\rho_\mathrm{f}\bar{u}^2_{\mathrm{f},0}}, \tag{6.32}$$

and the lift coefficient as

$$C_\mathrm{l} = \frac{2F_{\mathrm{h},y}}{0.041\rho_\mathrm{f}\bar{u}^2_{\mathrm{f},0}}. \tag{6.33}$$

Table 6.2 lists the resulting drag and lift coefficients for different resolutions N and smooth boundary sizes ε along with the range provided by Schäfer et al. [172]. For all ε, it is noticeable that the higher N is, the better the HLBM results fit the range of Schäfer et al. [172]. In general there is good agreement, but for $N = 1$ there are large discrepancies, suggesting underresolution. Looking at the drag coefficient, this is particularly noticeable for high ε, as it increases the object size depending on the lattice spacing. This highlights the dependence of the results on ε, as we also find that using $\varepsilon = 2\Delta x$ leads to an overestimation of the drag and lift coefficients. Drag is even overestimated with $\varepsilon = 1\Delta x$, suggesting that $\varepsilon < \Delta x$ gives the best results.

To visualize the change over resolutions, Figure 6.3 shows the relative error in the L^2 norm [90] of the drag (left) and lift (right) coefficients versus N for different ε. For comparison, lines for the experimental orders of convergence (EOC) of 1 and 2 are also included. It is evident that the error of the baseline resolution of $N = 5$ decreases with increasing N. It can also be seen that the trend follows EOC $= 2$ over all ε and for both drag and lift coefficients, suggesting quadratic convergence.

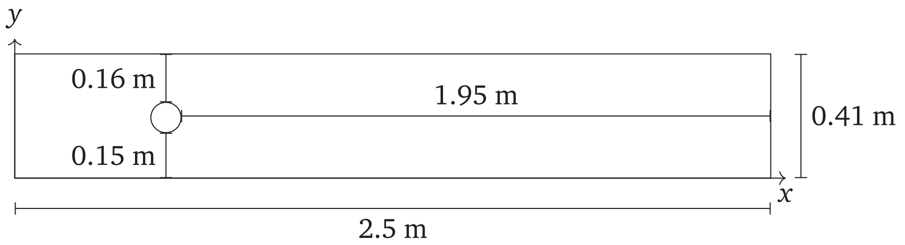


Figure 6.2: *Representation of the simulation setup for flow around a cylinder as a projection onto the x-y plane.*

Table 6.2: *Drag and lift coefficients computed in [67] for a flow around a cylinder.*

	Drag Coefficient C_d					**Lift Coefficient C_l** $(\times 10^{-2})$				
	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$
$\varepsilon = 2\Delta x$	9.83	7.57	7.05	6.81	6.68	0.00	1.04	1.11	1.11	1.09
$\varepsilon = 1\Delta x$	8.15	6.90	6.63	6.51	6.43	0.00	0.66	0.81	0.87	0.90
$\varepsilon = 0.5\Delta x$	6.93	6.44	6.35	6.30	6.27	0.00	0.77	0.88	0.91	0.92
$\varepsilon = 0\Delta x$	6.90	6.38	6.24	6.22	6.19	0.00	0.53	0.66	0.73	0.76
Schäfer et al. [172]	6.05 – 6.25					0.80 – 1.00				

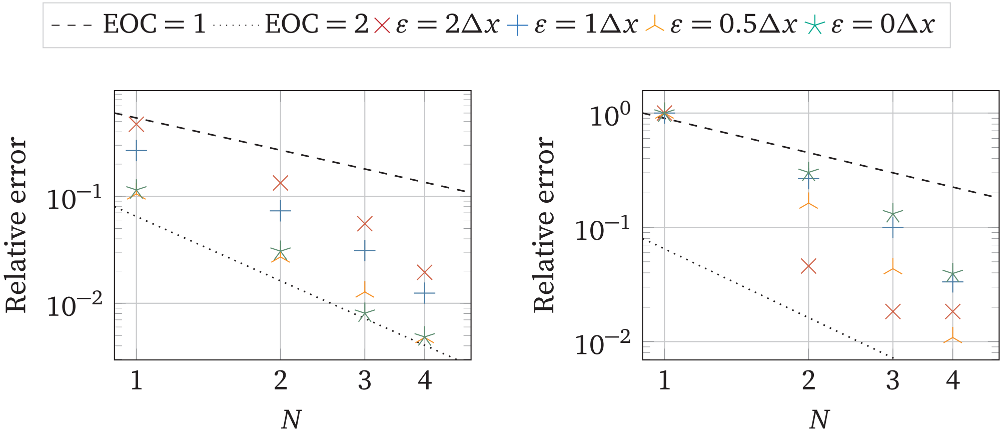


Figure 6.3: *Relative error of the drag (**left**) and lift coefficients (**right**) computed in [67] in L^2 norm versus the resolution N.*

6.4.2 Settling Sphere

As a basic validation of moving particles, Trunk et al. [69] compare the simulation results of a simple single settling sphere with the experimental results of ten Cate et al. [174]. The corresponding simulation setup is illustrated in Figure 6.4. Here, a sphere with a diameter of $D_s = 0.015$ m is placed in the center of a cubic container with dimensions of 0.1 m × 0.1 m × 0.16 m at a height of 0.12 m. The spherical particle has a density of $\rho_p = 1120\ \mathrm{kg/m^3}$, while the density and viscosity of the surrounding fluid vary. The values used in the different cases are given in Table 6.3.

In the HLBM simulations, the walls are modeled using a no-slip halfway bounce-back condition [91]. Originally, Trunk et al. [69] considered several forc-

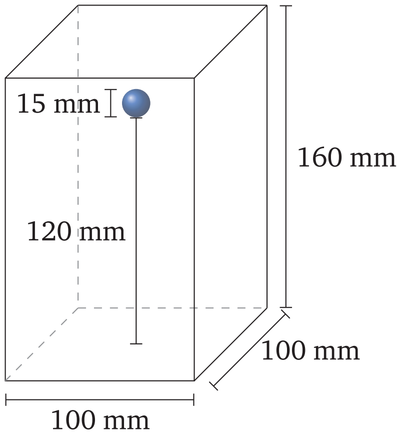


Figure 6.4: *Simulation setup of a single settling sphere according to Trunk et al. [69] and ten Cate et al. [174].*

Table 6.3: *Values considered for the computation of a single settling sphere in [69].*

Case	Fluid Density ρ_f in kg/m^3	Dynamic Viscosity η in Pa s	Grid Spacing Δx in mm	Time Step Size Δt in ms
1	970	0.373	1.671	0.3891
2	965	0.212	1.645	0.2410
3	962	0.113	1.610	0.1526
4	960	0.058	1.559	0.1010

ing and momentum exchange schemes as well as a wide range of discretization parameters. Since these studies found that a combination of the above MEA and the adapted EDM produced the most accurate results, however, the following studies using HLBM and the remainder of this paper also exclusively consider these as the superior reported schemes. We also limit the grid spacing and time step size to the values given in Table 6.3.

Figure 6.5 compares the HLBM simulation results with the experimental data for the cases mentioned in Table 6.3 by plotting the instantaneous particle velocity over time. The simulation and experimental data agree very well, hence, allow similar conclusions to be drawn. From case 1 to case 4, the density ratio increases, whereas the viscosity decreases. Therefore, the maximum settling velocity and acceleration also increase, resulting in shorter settling times as the particle contacts the bottom of the container more quickly.

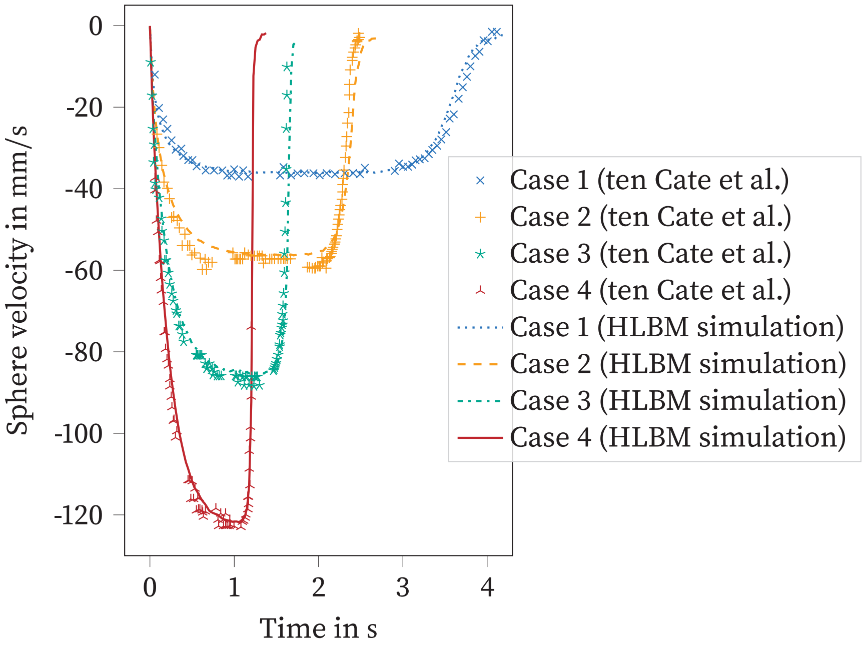


Figure 6.5: *Comparison of HLBM simulation results with experimental measurements by ten Cate et al. [174].*

Trunk et al. [69] also compare the predicted terminal velocities and drag coefficients of settling spheres for varying Reynolds numbers *Re* with well-established drag correlations and conclude that the lattice velocity should not exceed 0.04, while a lattice velocity of 0.01 is considered most desirable. Furthermore, the authors report linear convergence, which differs from the convergence observed by [67], see Section 6.4.1, which is likely due to a less significant influence of the surrounding walls and the consideration of moving curved boundaries for which a staircase approximation is used [72, 90, 101]

6.4.3 Shape-dependent Settling of Single Particles

After the above validation, Trunk et al. [68] employed HLBM to derive a shape-dependent drag correlation for single particle settling. To this end, the authors first identified three independent shape parameters, namely, the elongation κ_{el}, the roundness κ_{rnd} [175], and the Hofmann shape entropy λ_{H} [176], using

Pearson correlation coefficients [177]. These shape parameters are defined as

$$\kappa_{\mathrm{el}} = a_{\mathrm{I}}/a_{\mathrm{L}}, \tag{6.34}$$

$$\kappa_{\mathrm{rnd}} = \frac{V_{\mathrm{p}}}{A_{\mathrm{p}}(8a_{\mathrm{L}}a_{\mathrm{I}}a_{\mathrm{S}})^{1/3}}, \tag{6.35}$$

$$\lambda_{\mathrm{H}} = -\frac{\tilde{a}_{\mathrm{S}} \ln \tilde{a}_{\mathrm{S}} + \tilde{a}_{\mathrm{I}} \ln \tilde{a}_{\mathrm{I}} + \tilde{a}_{\mathrm{L}} \ln \tilde{a}_{\mathrm{L}}}{\ln 3}, \tag{6.36}$$

where V_{p} and A_{p} are the particle's volume and surface. a_{S}, a_{I}, and a_{L} correspond to the shortest, intermediate, and longest particle's half-axis, respectively, which are normalized as $\tilde{a}_j = a_j/(a_{\mathrm{S}} + a_{\mathrm{I}} + a_{\mathrm{L}})$ for $j \in \{\mathrm{S}, \mathrm{I}, \mathrm{L}\}$.

To obtain the necessary data from simulations, the authors separately considered 200 superellipsoids in a domain similar to that in Figure 6.4. However, the width and depth were set to 7 mm, the height to 17.5 mm, and the particles were placed in the middle of the container with a distance of 9.554×10^{-1} mm from the top wall. The surrounding fluid had a density of $\rho_{\mathrm{f}} = 998\,\mathrm{kg/m^3}$ and a viscosity of $\eta = 9.98 \times 10^{-4}\,\mathrm{Pa\,s}$. In addition, the standard gravity $g = 9.81\,\mathrm{m/s^2}$ was applied and the following correlation of drag coefficients for single particle settling with a coefficient of restitution of 0.96 was reported [68]:

$$\begin{aligned} C_{\mathrm{d}} = &-0.386\kappa_{\mathrm{el}} - 0.496\kappa_{\mathrm{rnd}} - 0.575Re + 0.097\kappa_{\mathrm{el}}\kappa_{\mathrm{rnd}} - 0.322\kappa_{\mathrm{el}}Re \\ &+ 0.0497\kappa_{\mathrm{rnd}}^2 - 0.128\kappa_{\mathrm{rnd}}\lambda_{\mathrm{H}} + 0.106\kappa_{\mathrm{rnd}}Re - 0.296\lambda_{\mathrm{H}}Re + 0.161Re^2, \end{aligned} \tag{6.37}$$

where the Reynolds number $Re = \rho_{\mathrm{f}} D_{\mathrm{eq}} u_{\infty}/\eta$ was obtained from the diameter of a volume-equivalent sphere D_{eq} and the terminal velocity u_{∞}.

Utilizing Eq. (6.37), it is now feasible to include shape dependence on the subgrid scale and in Euler—Euler particle simulations to perform investigations at a process scale, which can also benefit from the above straightforward LBM parallelization. Particle simulations coupled with LBM have been demonstrated in Euler–Euler simulations of, for example, a bifurcation [159] and a high-gradient magnetic separation [27]. In addition, LBM was coupled with subgrid scale particle simulations to investigate the nasal cavity [97].

6.4.4 Cylinder-wall Impact

To include dense suspensions in the possible applications of HLBM, Marquardt et al. [71] introduced and validated the discrete contact model by considering a cylinder-wall impact according to Park [113]. A sketch of the corresponding setup

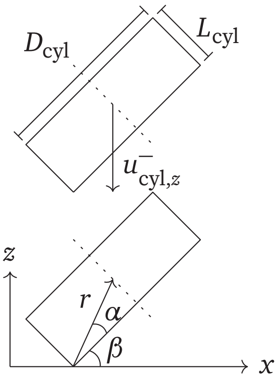


Figure 6.6: *Cylinder-wall impact simulation setup according to [71].*

is shown in Figure 6.6. It displays the cylinder of density $\rho_{\text{cyl}} \approx 1164\ \text{kg/m}^3$, diameter $D_{\text{cyl}} = 8$ mm, and length $L_{\text{cyl}} = 5.3$ mm. The cylinder settles with a constant initial velocity $u^-_{\text{cyl}} = -1$ m/s until it contacts the bottom wall, which means that there is no interaction between the fluid and the particle. The distance between the point of impact and the center of the cylinder is given by $r = \sqrt{(D_{\text{cyl}}/2)^2 + L^2_{\text{cyl}}/4}$. The angle α is between the line from the center of the cylinder to the point of impact and the face of the cylinder, while the impact angle β is between the bottom of the cylinder and the flat wall.

The expected rebound angular velocity is given by [113]

$$\omega^+_{\text{cyl},y} = \frac{m u^-_{\text{cyl},z}(1+e) r \cos(\alpha+\beta)}{I_{\text{yy}} + m r^2 \cos^2(\alpha+\beta)}, \tag{6.38}$$

where m is the mass of the cylinder, I_{yy} is the moment of inertia with respect to the y-axis, and $e = u^+_{\text{cyl},z}/u^-_{\text{cyl},z}$ is the signed coefficient of restitution with the rebound velocity [113]

$$u^+_{\text{cyl},z} = \omega^+_{\text{cyl},y}\, r \cos(\alpha+\theta) - e u^-_{\text{cyl},z}. \tag{6.39}$$

In their simulations, the authors considered $e = -0.85$, $k = 2/\sqrt{\pi}$, and resolved the cylinder diameter with $N = 4$ cells for the mesh-based contact detection, while varying the contact region resolution N_{c}. Figure 6.7 compares the computed rebound velocity (left) and rebound angular velocity (right) with the above solutions. In general, there is a strong dependence on the contact resolution N_{c}, since for $N_{\text{c}} = 2$, the contact is clearly underresolved, leading to high discrepancies. With increasing N_{c}, however, the accuracy improves, so

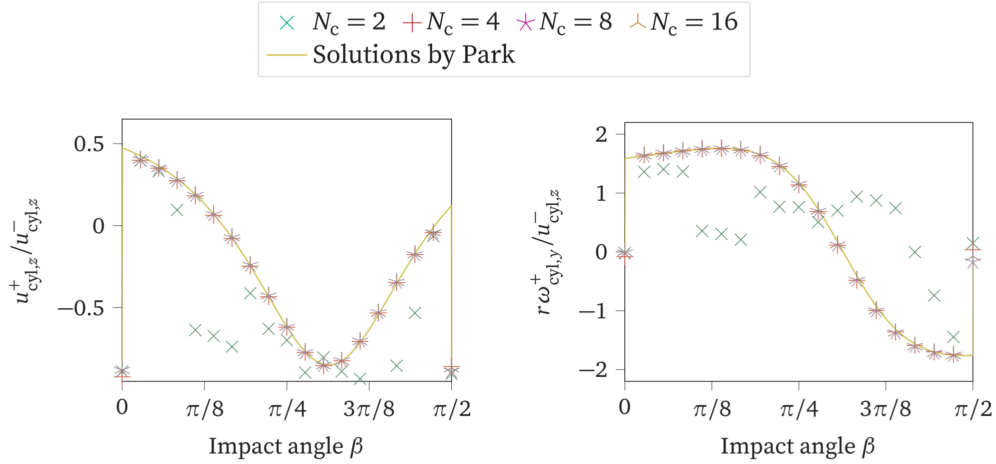


Figure 6.7: *Comparison of the simulation results of [71] with the solutions by Park [113], see Equations (6.38) and (6.39). The rebound velocity is shown on the left and the rebound angular velocity is shown on the right.*

that already for $N_c = 4$ a good agreement is visible, but the authors still suggest to use $N_c \geq 8$, as this also leads to more accurate contact forces.

6.4.5 Sphere Rebound

For the validation of four-way coupled HLBM simulations, Marquardt et al. [71] considered the rebound of spherical particles in a viscous fluid according to Li [114]. In this case, a sphere was placed in the middle of a cubic box similar to the setup outlined in Figure 6.4. However, the container measured 42 mm × 42 mm × 50 mm and the sphere with a density of $\rho_p = 7780\,\mathrm{kg/m^3}$ was placed at various distances, 5.5 mm, 19.6 mm, and 35.7 mm, from the bottom wall. Furthermore, the authors set the Young's modulus of the particle to 20 GPa and that of the wall to 3 GPa, while the Poisson's ratios were 0.33 and 0.24, respectively. The fluid had a density of $\rho_f = 1230\,\mathrm{kg/m^3}$ and a viscosity of $\eta = 50.2\,\mathrm{Pa\,s}$.

Again, the surrounding walls were modeled using a halfway bounce-back condition [91]. The simulations used a lattice spacing of $\Delta x = 0.3$ mm and time step size of $\Delta t = 5\,\mu\mathrm{s}$, while the contact resolution $N_c = 16$, constant $k = 4/(3\sqrt{\pi})$, and the damping factor $c = 0.12$ s/m were chosen. To decouple the particle and fluid calculations, the authors introduced smaller time steps of $\Delta t\,/\,1000$ for solving the equations of motion and calculating the contact forces. They also increased the particle size by $\Delta x\ /\ 7$ for the contact treatment only, as otherwise the viscous forces would become unphysically high due to the assumption of an incompressible fluid.

Figure 6.8 shows the distance of the sphere surface to the bottom wall over time for the three setups mentioned above. The lines correspond to the HLBM simulation results [71] and the points correspond to the experimental data [114]. In both cases, a similar trend can be seen: The higher the initial position is, the steeper is the curve, which means that the maximum velocities of the spheres increase with the distance. Also, the spherical particle rebounds more often when the distance increases, which is caused by the higher amount of potential energy that is converted into kinetic energy in these cases.

Comparing the HLBM simulation with the experimental data, we observe a good agreement, especially for the initial distances of 5.5 mm and 19.6 mm. For an initial distance of 35.7 mm, however, there is a comparatively higher discrepancy after the first rebound, which is probably due to the use of a constant damping factor c, which, according to Equation 6.19, should actually depend on the initial relative velocity, which obviously changes among the cases, as already noticed before.

6.4.6 Wall-flow Filters

Hafen et al. [86–88, 129] considered the application to wall-flow filters, which is important because the exhaust from combustion engines contains particulate matter, which is a potential health hazard and therefore needs to be captured. To study the application numerically, the authors consider a single channel of a wall-flow filter as shown in Figure 6.9. Here, the walls and particles are made of porous media, which is modeled with HLBM. A constant velocity was set at the inlets and a constant pressure was set at the outlets. Periodic boundaries were applied perpendicular to the flow direction to represent real wall flow filters, which consist of hundreds of individual channels.

First, the flow without any particles was considered [86], showing super-linear grid convergence and good agreement with a reference solution [178]. It was also found that for non-spherical particles, lift forces are not the primary detachment forces, while the stopping distance is so long that back-end deposition is inevitable [129]. Fragment detachment occurs in successive stages, as the first row of fragments experiences the greatest hydrodynamic forces while shielding the following fragments, thus reducing the hydrodynamic forces and the likelihood of detachment for those fragments [87].

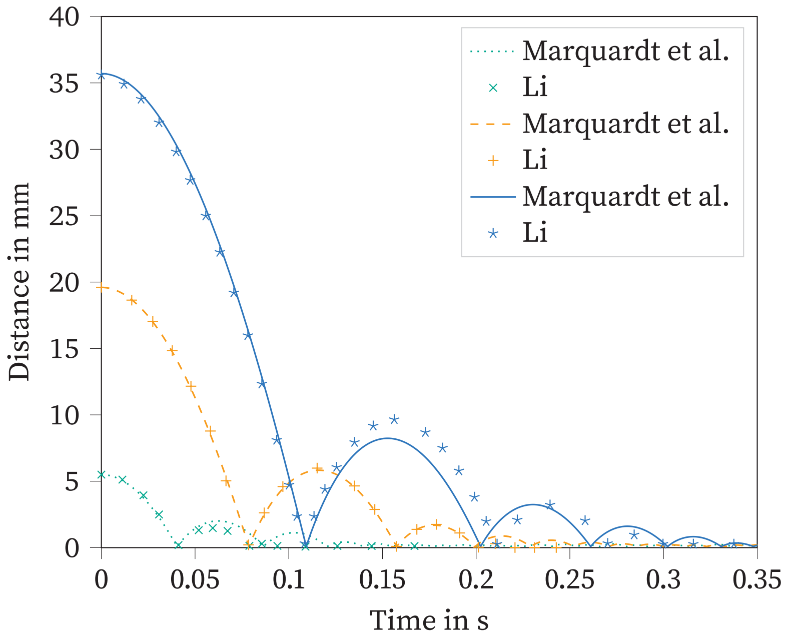


Figure 6.8: *Plot of the shortest distance between the surface of a settling sphere and the bottom wall, comparing HLBM simulations by Marquardt et al. [71] with experimental data by Li [114].*

Finally, [88] dealt with particle detachment and plug formation as shown in Figure 6.9. The results indicate that a straightforward mean density methodology serves as a consistent method for measuring both the extent of a channel plug and its mean density, which closely matches values documented in the existing literature [179, 180]. The study also shows that the topology of the fragment layer affects the final plug structure and, more significantly, the pressure drop. Larger layer heights caused by longer loading times or higher particle concentrations result in increased pressure drops in the plug state. The same effect is observed for larger fragment dimensions associated with a lower soot-to-ash ratio. The more particles are present after soot oxidation, the higher is the pressure drop. A transition to a particulate plug becomes advantageous when the fragmented layer limits the free substrate surface area in contact with the flow. Increasing the density of the particulate matter has been found to decrease the pressure drop because the higher inertia leads to a greater compactness of the plug. Following this reasoning, the authors mention that the use of additives to decrease the stiffness of the particles would also increase the compactness

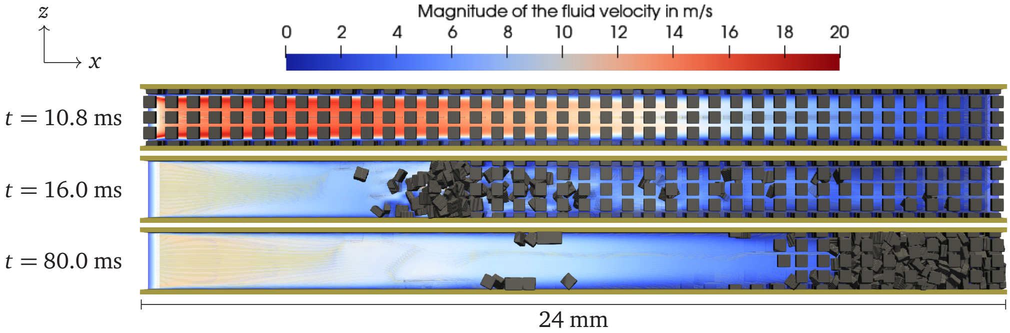


Figure 6.9: *Chronological evolution of detachment, transport, and plug formation in a wall-flow filter as a sequential series [88].*

and therefore reduce the pressure drop. Removal of the first few rows of fragments significantly reduces the pressure drop, while adhesive forces noticeably affect detachment at the threshold of the first row only. When examining the relationship between gas velocity and momentum loss in the filter substrate, it was found that the dependence on pressure drop in the plugged state is less significant.

6.4.7 Hindered Settling

In order to improve the performance of HLBM, Marquardt et al. [72] introduced a particle decomposition and validated it using hindered settling [77]. However, the authors solely considered two-way coupling and found that especially at high particle volume fractions, four-way coupling is necessary to obtain accurate results. For this reason, the implementation was extended and a parallelization strategy for the discrete contact model was introduced, leading to further studies of the hindered settling phenomena [73].

The authors considered both spheres of $D_\mathrm{s} = 3$ mm in diameter and volume-equivalent cubes in a triple periodic domain with an edge length of $12D_\mathrm{s}$ that is filled with a fluid having the density of $\rho_\mathrm{f} = 1000\ \mathrm{kg/m^3}$. In both cases, the particles with a density of $\rho_\mathrm{p} = 3300\ \mathrm{kg/m^3}$ initially rest at exactly the same randomly generated positions and the standard gravity $g = 9.80665\ \mathrm{m/s^2}$ is applied. The fluid viscosity is set depending on the considered Archimedes number $Ar \in \{500, 1000, 2000\}$, which is defined as

$$Ar = \frac{g D_\mathrm{s}^3 \rho_\mathrm{f}(\rho_\mathrm{p} - \rho_\mathrm{f})}{\eta^2}. \tag{6.40}$$

In addition, the authors consider particle volume fractions ϕ_p between 10% and 30%, while a consistent lattice relaxation time $\tau = 0.55$ is used. Furthermore, the authors vary the number of cells N per sphere diameter D_s. A contact resolution of 8, $k = 4/(3\sqrt{\pi})$, $e = -0.926$, and $u_k = 0.001\mathrm{m/s}$ were used.

Examining the results of the swarm of settling spheres with $N = 18$, the authors find a good agreement with literature even without an explicit contact model for low particle volume fractions. However, as the particle volume fraction increases, the need for an explicit contact model becomes more apparent. The authors point out that for particle volume fractions $\phi_\mathrm{p} > 0.3$, a lubrication force correction model must be introduced to accurately capture the settling behavior.

Figure 6.10 shows the velocity field around the swarm of spheres (left) and cubes (right) at different times for a resolution of $N = 18$, particle volume fraction of $\phi_\mathrm{p} \approx 0.3$, and Archimedes number of $Ar = 2000$. While voids with high fluid velocities form between the spheres, the fluid velocity is much more evenly distributed when cubes are considered instead.

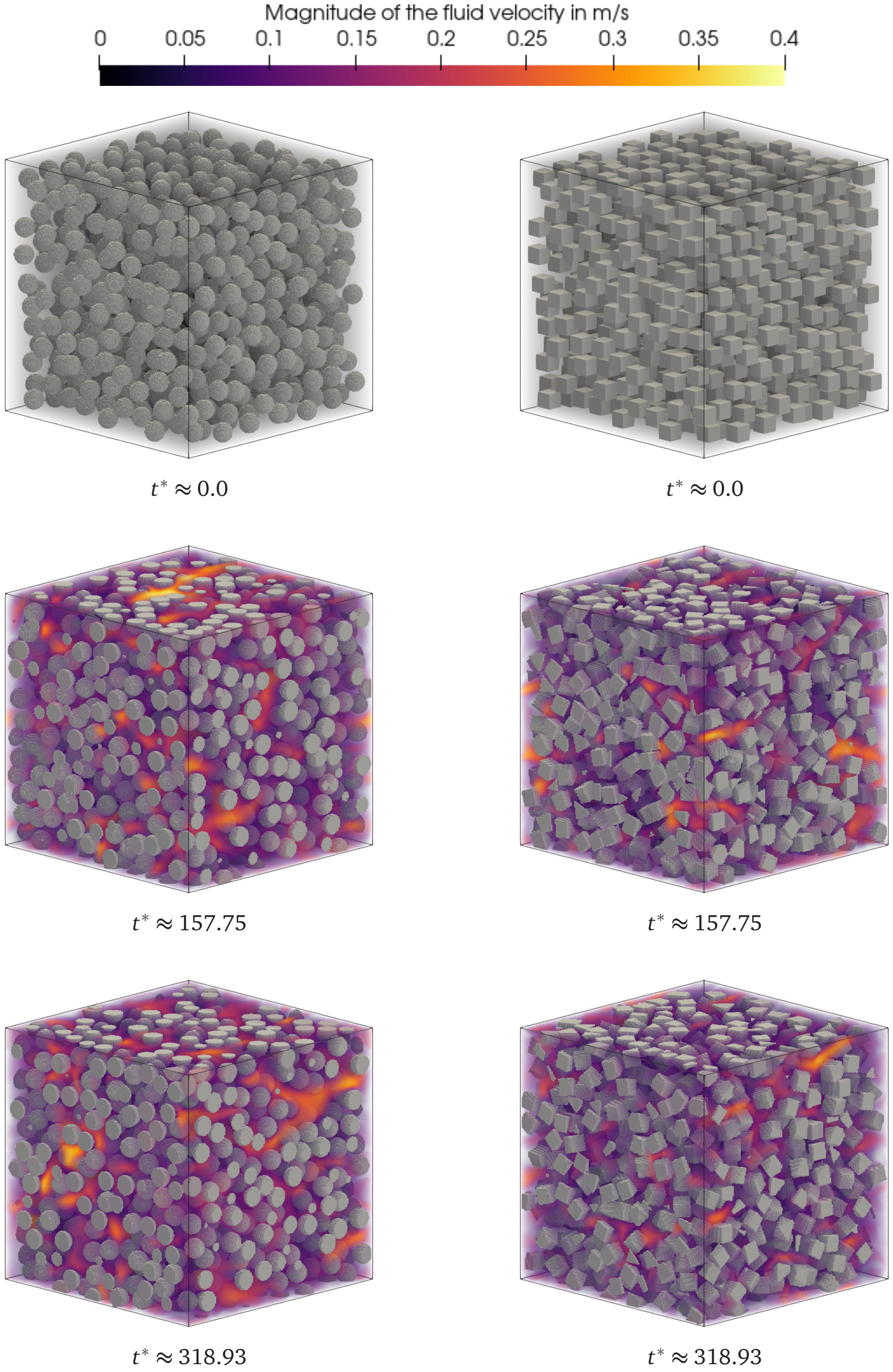
This is also reflected by the average settling velocity of the particles, which is plotted over the particle volume fraction ϕ_p in Figure 6.11. For the normalization, the average settling velocity $\bar{u}_\mathrm{p}$ is divided by the terminal settling velocity of a single sphere, which reads

$$u^* = \sqrt{\frac{4gD_\mathrm{s}}{3C_\mathrm{d}}\left(\frac{\rho_\mathrm{p} - \rho_\mathrm{f}}{\rho_\mathrm{f}}\right)}. \tag{6.41}$$

To calculate the drag coefficient, the authors use the correlation by Schiller and Neumann [99] given by

$$C_\mathrm{d} = \frac{24}{Re}\left(1 + 0.15Re^{0.687}\right). \tag{6.42}$$

Figure 6.11 shows that in all cases the spheres settle faster than the cubes on the average, i.e. the cubes are 13 to 26% slower than the spheres. The authors reason that this is partly due to the spheres tending to build clusters more easily, as observed above, leaving voids for the fluid to pass through, thus reducing the velocity difference at the particle surface and, hence, the drag. Spheres are also rotationally invariant, which is why the rotation of the particle has little effect on the fluid. In cubes, on the other hand, the rotation changes the cross-sectional area and has a pronounced effect on the fluid.



$t^* \approx 0.0$

$t^* \approx 0.0$

$t^* \approx 157.75$

$t^* \approx 157.75$

$t^* \approx 318.93$

$t^* \approx 318.93$

Figure 6.10: *Representation of the velocity field around settling spherical (**left**) and cubic (**right**) particles in a triple periodic domain at different normalized times t^*, an Archimedes number of 2000, and a particle volume fraction of about 30% [73].*

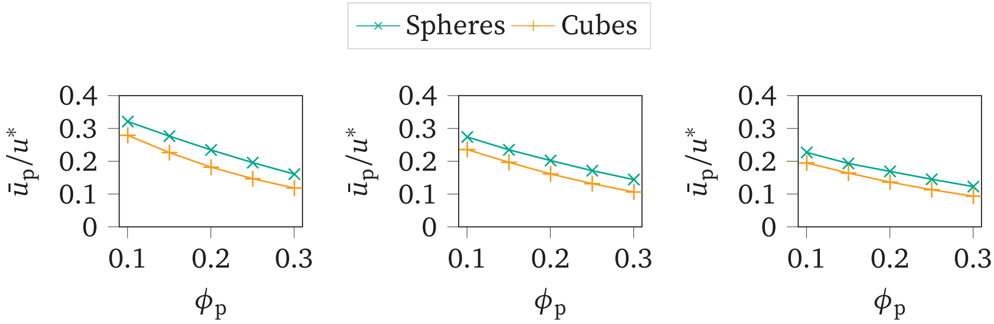


Figure 6.11: *Comparison of the normalized average settling velocity of spheres and cubes at $Ar = 500$ (**left**), $Ar = 1000$ (**middle**), and $Ar = 2000$ (**right**).*

6.4.8 Transport through a Cross-sectional Constriction

HLBM has been used to investigate the mechanical stress during transport through cross-sectional constrictions, as fully resolved particles are necessary to determine local forces on the surfaces [171]. In their study, the authors considered a setup, which is shown as a projection onto the x-y plane in Figure 6.12. At the inlet, the 500 mm long geometry starts with a cylinder that has a diameter of 50 mm and covers 70% of the length. After that, the geometry narrows down to 25 mm. The taper angle θ quantifies the rate at which the diameter of the pipe decreases along the conical transition. The non-Newtonian fluid is considered to be a Herschel–Bulkley fluid [181] with a density of 1100 kg/m^3, a flow index of 0.42, a consistency of 13.1 Pa s$^{0.42}$, and a yield stress of 82.7 Pa. The cubic particles initially occupy 30% of the volume in the section with the big diameter and have the same density as the fluid. Particle edge lengths of 6.4 and 8.1 mm are considered. The particles have Young's moduli of 30 kPa and 60 kPa. The walls have Young's moduli of 190 GPa and the same Poisson ratio of 0.3. The coefficient of restitution is set to 0.9, the coefficient of kinetic friction to 0.1, and the coefficient of static friction to 0.15. A time interval of 1.1 s is simulated. For the evaluation, the first 0.1 s are neglected. The walls are considered no-slip boundaries using the Bouzidi boundary approach [182]. A constant pressure is set at the outlet. At the inlet, a constant velocity profile from prior simulations is used. While the setup applied for these simulations is the same as above, but the velocity is set to the average velocity over the entire inlet, the length is shortened to 300 mm, and no particles are considered. When the simulation converged, the velocity profile at $x = 225$ mm was saved and afterwards used as

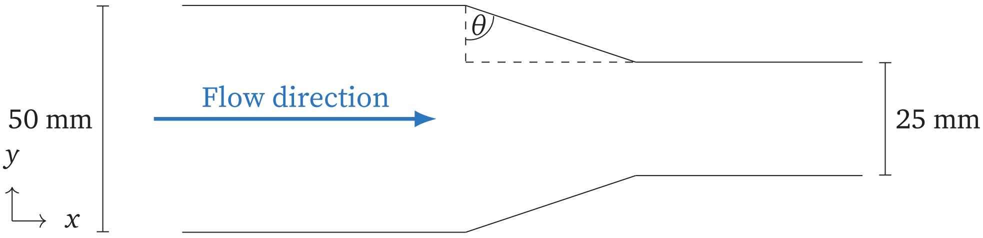


Figure 6.12: *Illustration of the setup studied in [171] as a projection onto the x-y plane, highlighting the diameter of the cylindrical pipe at the inlet and its narrowing towards the end, with a predefined taper angle θ.*

the boundary and initial condition for the following studies with particles.

Metzner–Reed Reynolds numbers [183] between 2 and 8 were studied and found have the greatest influence on the fluid- and contact-induced mechanical stress, along with particle properties, such as Young's modulus and size. The taper angle θ showed no significant influence. In other words, as the pipe narrows, fluid-induced mechanical stress increases primarily because the Metzner-Reed Reynolds number increases in the smaller diameter sections. Contact-induced mechanical stress increases due to higher relative velocities at the initial contacts.

Results of an example with an edge length of 8.1 mm, a particle Young's modulus of 60 kPa, and $\theta = 45°$ are given in Figure 6.13. Here, the average distance of the center of mass of the particles from the center of the pipe is shown over the length of the pipe (top left), where $x = 0$ m corresponds to the beginning of the taper. The number of particle-particle and particle-wall contacts (top right), fluid-induced stress (bottom left), and contact-induced stress (bottom right) are also plotted over the length of the pipe. The plot displays the median induced stresses as a line, accompanied by an area encompassing the central 90% of the data, thus excluding the 5% lowest and 5% highest stresses to identify trends while reducing the influence of outliers.

The results show that the particles move towards the center of the pipe due to the cross-sectional constriction. As a result, interparticle contacts increase due to rearrangements on the way towards the constriction, while they remain at a constant level afterwards. Particle-wall contacts occur most frequently at the beginning of the constriction and before. They become less likely in the small-diameter section. Looking at the bottom plots, the normal mechanical stress is greater than the tangential component. While the fluid-induced stresses become smaller after the constriction, the contact-induced normal stresses remain on a similar level. The contact-induced tangential stresses increase, which is likely due to higher relative velocities.

6.5 Discussion

6.5.1 Strengths and Weaknesses

HLBM has several strengths that make it a versatile tool in fluid dynamics simulations. A key strength is its monolithic approach, which allows for seamless integration of various fluid-solid interaction scenarios in a unified framework. HLBM enables the simultaneous simulation of pure fluid flow, porous media, and moving porous particles, thereby streamlining the modeling process.

Another major advantage is its ability to handle arbitrary shapes, allowing for the consideration of a wide range of particle geometries. Specifically, HLBM

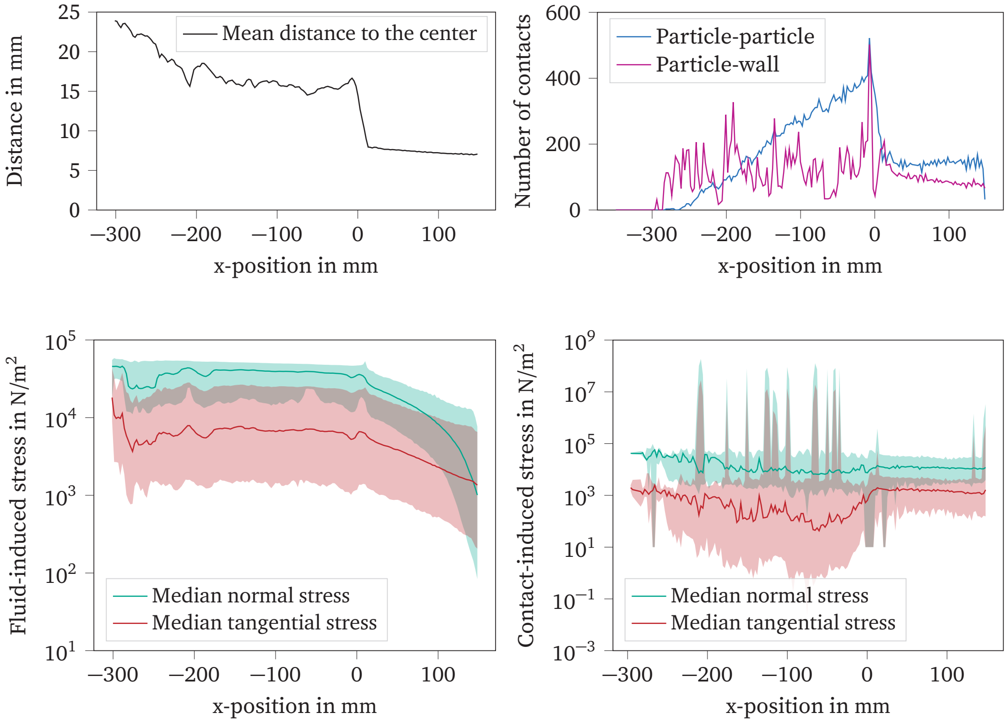


Figure 6.13: *Plot of the average distance to the center of the pipe (**top left**), number of contacts (**top right**), fluid-induced stress (**bottom left**), and contact-induced stress (**bottom right**) versus the length of the pipe.*

accommodates arbitrary shapes with two-way coupling [67, 68], while it efficiently handles convex shapes with four-way coupling [71, 73], using the discrete contact model mentioned before. It accurately resolves particle shapes, thereby facilitating the extraction of forces acting on these particles - an essential aspect in various scientific investigations.

In addition, HLBM can be easily parallelized, which enhances its computational efficiency. Leveraging the strengths of the LBM, HLBM takes advantage of parallel computing architectures to optimize performance and speed up simulations [72, 73]. Its robustness is underscored by extensive testing over the years, establishing a track record of reliability and accuracy in diverse applications, as outlined above. In particular, HLBM uniquely treats particles as porous media since the coupling to fluid is done using a locally confined permeability [86], a feature lacking in many alternative methods. This feature is particularly important in scenarios where the behavior of porous media significantly affects system dynamics [86–88, 129] and it distinguishes HLBM as a comprehensive modeling approach.

Since HLBM is based on LBM, however, it requires small time steps due to diffusive scaling, i.e. $\Delta t \propto \Delta x^2$. Compared to alternative methods, such as subgrid scale or Euler–Euler particle simulations, HLBM has relatively high computational requirements so far, typical of DNS approaches. In addition, HLBM is currently limited to modeling rigid particles [66, 67, 69], which restricts its applicability in scenarios involving deformable or flexible entities.

In summary, HLBM is characterized by its ability to accommodate diverse particle shapes, resolving particle shapes, its parallelizability, and its treatment of particles as porous media. However, the need for small time steps, computational cost, and limitation to rigid particles are important considerations when applying this method to fluid dynamics simulations.

6.5.2 Outlook on Future Development

The future trajectory of HLBM is promising and there are several key areas needing advancement and refinement.

In general, performance improvements should be addressed in future developments. Optimization of algorithms and computational strategies can lead to improved efficiency, resulting in faster simulations without compromising accuracy.

Big improvements are to be expected by the implementation of hybrid parallelization techniques. The integration of hybrid parallelization techniques would significantly reduce communication overhead, a critical aspect in large-scale simulations. By efficiently distributing computational tasks across different computing resources while minimizing inter-node communication, this approach could significantly accelerate simulations, thereby extending the applicability of the method to more complex and larger systems.

Another key advancement would be the integration of graphics processing unit (GPU) support. Leveraging the computational power of GPUs could significantly accelerate HLBM simulations by taking advantage of their parallel processing capabilities. This improvement has the potential to revolutionize the efficiency of the method, allowing for faster and more scalable simulations across platforms.

Further advances in HLBM could also include extending the discrete contact model to include concave particles. Updating the contact detection mechanism to recognize and handle multiple contacts between pairs of particles would greatly increase the versatility of the method. This is because contact detection is currently unable to distinguish between connected and isolated parts of contacts. However, this could be added easily by running a second iteration over the rough detection and then splitting it into the separate contacts. Enabling HLBM to accurately model interactions involving concave particles is a critical step towards simulating more complicated real-world scenarios, thereby broadening its scope of applicability.

Another important future development is the addition of a lubrication force correction model. The introduction of this correction model could enable HLBM to accurately capture and account for lubricating forces between particles in close proximity. This addition would increase the accuracy of the method in scenarios where such forces play a critical role, thereby improving its predictive capabilities in simulating particle dynamics under varying conditions.

Deformable particles may potentially be supported by an integration of other LBM-based approaches that allow to solve nonlinear, transient, large displacement solid mechanics [184, 185].

In summary, HLBM promises to meet upcoming scientific and industrial challenges. Performance improvements, hybrid parallelization, GPU support, extension of contact models to handle concave particles, integration of lubricating force corrections, and allowing for deformable particles are key challenges that would strengthen HLBM's capabilities and enable more accurate, efficient, and versatile simulations in a wide range of applications.

6.6 Summary and Conclusions

In summary, HLBM is a versatile computational framework that is also widely used for the study of particulate flows. This paper has provided a comprehensive overview of HLBM, elucidating its fundamental principles and presenting its diverse applications in particle-laden flows as documented in the existing literature.

Notable strengths of HLBM include its monolithic nature, which allows for the seamless simulation of diverse fluid-solid interactions, and its ability to handle arbitrary particle shapes, even irregular and concave geometries, while accurately capturing local surface interactions. In addition, its parallel scheme based on LBM ensures high computational efficiency, making it suitable for large-scale simulations. However, being LBM-based comes with the drawback of requiring small time steps.

Future development needs focus on the underlying models and performance improvements. The discrete contact model needs to be extended by a lubrication force correction model to improve accuracy at particle volume fractions $\phi_\mathrm{p} > 0.3$ and to support concave shapes. Allowing for deformable particles would also be important to extend HLBM's applicability. DNS, including HLBM, are computationally expensive. Increasing the performance e.g. by introducing hybrid parallelization techniques and GPU support would speed up simulations and also allow a much larger number of surface resolved particles to be considered.

These developments promise to greatly expand the capabilities of HLBM and increase its usefulness in solving complex real-world problems.

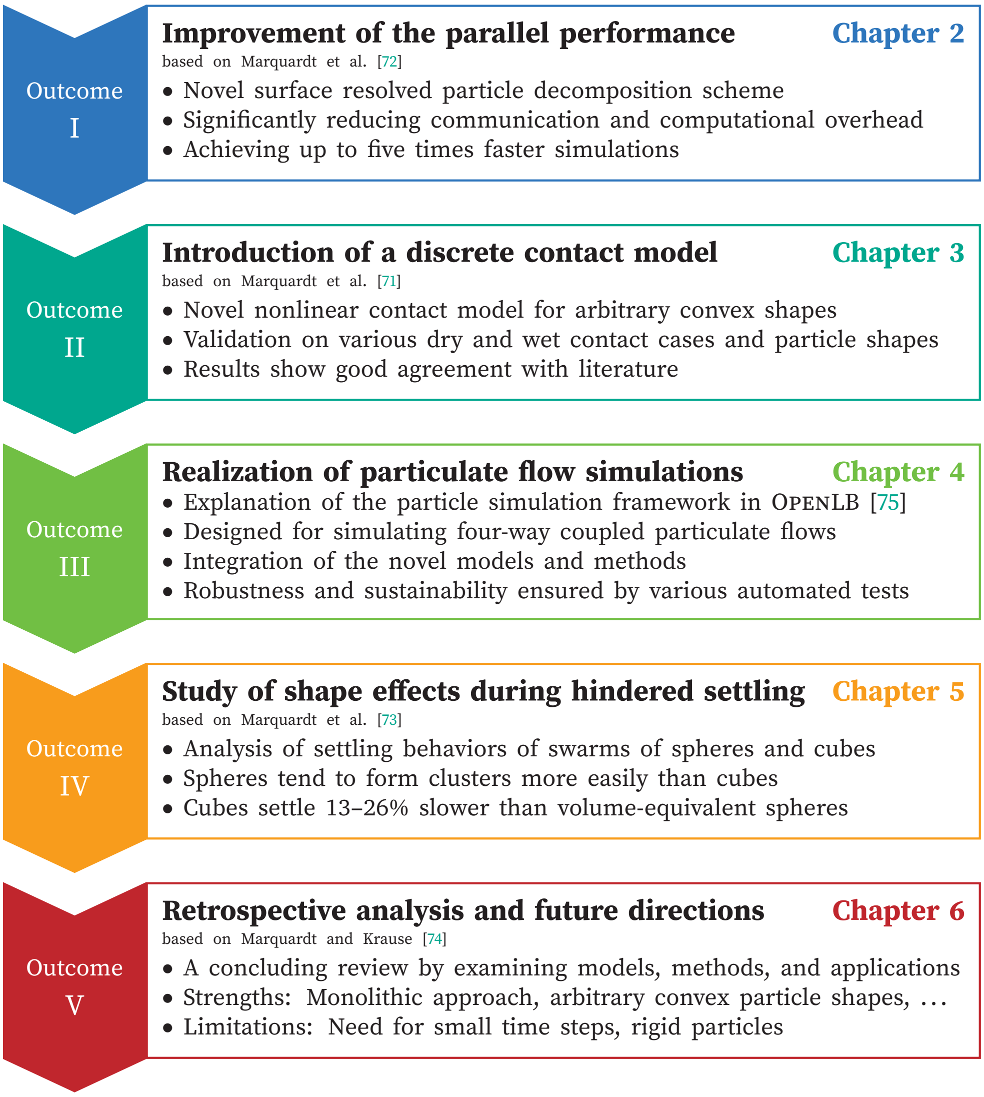
7 Conclusion and Outlook

7.1 Conclusion

This work successfully achieves the main objective stated in the Chapter 1 by introducing and rigorously validating a novel LBM-based model for the DNS of particulate flows, focusing on non-colloidal suspensions and allowing for high particle volume fractions of up to 30%. To this end, the parallel performance of surface resolved particle simulations was improved by a particle decomposition scheme (Subgoal 1) and four-way coupled simulations of convex particles were made possible by a discrete contact model (Subgoal 2). These new developments are part of a flexible, reusable, freely available, and easily extensible software implementation (Subgoal 3), which was used for detailed studies of the hindered settling of spheres and cubes to demonstrate the capabilities of the above and to investigate suspension dynamics involving cubic particles (Subgoal 4).

Following the thesis structure explained in Section 1.3, a concise visual summary of the contributions of this thesis is provided in Figure 7.1, while a detailed description of each contribution is given below:

Outcome I — Improvement of the parallel performance A surface resolved particle decomposition scheme compatible with HLBM was introduced in Chapter 2. It addresses the communication and computational overhead by restricting communication to a defined neighborhood of each PU, rather than having all PUs communicate with each other. Each PU is assigned a specific subset of par-



Outcome I

Improvement of the parallel performance **Chapter 2**

based on Marquardt et al. [72]

- Novel surface resolved particle decomposition scheme
- Significantly reducing communication and computational overhead
- Achieving up to five times faster simulations

Outcome II

Introduction of a discrete contact model **Chapter 3**

based on Marquardt et al. [71]

- Novel nonlinear contact model for arbitrary convex shapes
- Validation on various dry and wet contact cases and particle shapes
- Results show good agreement with literature

Outcome III

Realization of particulate flow simulations **Chapter 4**

- Explanation of the particle simulation framework in OpenLB [75]
- Designed for simulating four-way coupled particulate flows
- Integration of the novel models and methods
- Robustness and sustainability ensured by various automated tests

Outcome IV

Study of shape effects during hindered settling **Chapter 5**

based on Marquardt et al. [73]

- Analysis of settling behaviors of swarms of spheres and cubes
- Spheres tend to form clusters more easily than cubes
- Cubes settle 13–26% slower than volume-equivalent spheres

Outcome V

Retrospective analysis and future directions **Chapter 6**

based on Marquardt and Krause [74]

- A concluding review by examining models, methods, and applications
- Strengths: Monolithic approach, arbitrary convex particle shapes, …
- Limitations: Need for small time steps, rigid particles

Figure 7.1: *Overview of the key findings and contributions of each chapter of this thesis, the presentation follows the outline of the thesis structure in Figure 1.1.*

ticles to handle the corresponding purely particle-related computations. This strategy reduces the total number of inter-process communications because fewer PUs need to interact directly, resulting in less idle time. Although this approach requires more dedicated communication steps within each defined neighborhood, it significantly improves the scalability of the simulations. Indeed, the overall simulation performance increased by up to a factor of five in the setups evaluated, underscoring the importance of efficient communication and workload distribution. As a result, simulations with a significantly larger number of surface resolved particles are now much more feasible.

Outcome II — Introduction of a discrete contact model In a second contribution, discussed in Chapter 3, a contact treatment was developed that is compatible with HLBM. The underlying nonlinear contact model takes into account both elastic deformation and viscous damping aspects of interacting solid objects and considers both normal and tangential forces. The mesh-based approach discretizes the overlap region using a uniform mesh to calculate the overlap volume, contact point, contact normal, indentation depth, and contact force. This contact treatment has been rigorously validated using analytical solutions of contact forces for several dry contact scenarios, including contacts between two spheres, a sphere and a half-space, a cylinder and a half-space, and two crossed cylinders. To further evaluate its performance in dynamic situations, the case of cylinder-wall impact was considered. The results show excellent agreement with the analytical solutions, as the simulated dry contact cases closely match the theoretical predictions. In addition to the dry contact validation, the contact treatment has been evaluated using a wet contact benchmark, which involves a sphere rebounding from a wall in the presence of a viscous fluid. The simulated results show good agreement with the established benchmark, further demonstrating the robustness and versatility of the developed contact treatment and its compatibility with HLBM, thus enabling four-way coupled particulate flow simulations.

Outcome III — Realization of particulate flow simulations Chapter 4 presented a unified modular particle framework designed for four-way coupled particulate flow simulations as implemented in the open source software OpenLB [75], providing both an abstract overview and detailed implementation and usage details. The framework’s flexible architecture addresses the challenges of efficiently solving equations of motion for discrete particles in various physical scenarios by separating data and operations, while making it easily extensible without modifying existing implementations. It features a particle object that holds data such as position, velocity, and other properties, while operations such as solving the equations of motion are performed independently. The modular particle data allows data to be added or removed based on simulation needs, allowing it to be used for fully resolved and SGS particle simulations. In addition, the framework incorporates novel models and methods presented in this thesis. The chapter also highlights several measures taken to ensure software quality, including automated unit and benchmark testing. The particle framework is a significant and sustainable contribution because it provides a robust, flexible, and freely available tool for simulating discrete particle systems, enabling its use in future studies.

Outcome IV — Study of shape effects during hindered settling The application to hindered settling detailed in Chapter 5 emphasizes the development and successful implementation of a novel direct numerical simulation approach for complex particle systems, accommodating complex shapes and over a thousand of surface resolved particles. The studies performed confirm the grid independence and linear convergence of the method, ensuring the reliability and accuracy of the results. The results also highlight the importance of an explicit contact model in maintaining simulation accuracy, especially in high particle volume fraction and low Archimedes number scenarios. The final investigations of this chapter compare the settling behavior of swarms of spheres and cubes for Archimedes numbers between 500 and 2000 and particle volume fractions between 10 and 30%, leading to novel insights that cubes settle 13 to 26% slower than swarms of volume-equivalent spheres, while spheres exhibit a higher tendency to cluster. The findings indicate a strong shape dependence and suggest that shape plays an important role in separation processes and that a deeper understanding of shape-dependent suspension dynamics will lead to more accurate predictions and improved control of various industrial and environmental systems involving suspensions.

Outcome V — Retrospective analysis and future directions The final contribution is a comprehensive review of the LBM-based approach, presented in Chapter 6, providing an overview of its basic concepts and presenting its diverse applications in the study of particle-laden flows, including suspensions and aerosols, as reported in various studies. This review identifies the strengths and current limitations of the model. The monolithic approach of the model is a major advantage, allowing the seamless simulation of different scenarios involving fluid-solid interactions. This approach streamlines the modeling process by allowing the simultaneous simulation of pure fluid flow, porous media, and moving porous particles. Another key strength is the ability to handle arbitrary convex particle shapes in four-way coupled simulations and arbitrary shapes in two-way coupled simulations, while accurately representing local surface interactions. In addition, the parallel design, rooted in LBM, ensures superior computational efficiency, making it suitable for large-scale simulations. The model also has the unique ability to treat particles as porous media, a feature lacking in many alternative methods. This feature is particularly beneficial in scenarios where the behavior of porous media significantly affects system dynamics. However, because the model is based on LBM, it shares the disadvantage of requiring small time steps due to diffusive scaling, which results in relatively high computational requirements. These requirements are typically higher for DNS than for alternative methods such as SGS or Euler–Euler particle simulations. In addition, the model is currently restricted to modeling rigid particles, which limits its applicability in scenarios involving deformable or flexible entities. As a result, future development needs were identified, such as performance improvements through the integration of hybrid parallelization techniques and graphics processing unit (GPU), the extension of the discrete contact model to concave shapes, a dedicated lubrication force correction model, and the extension of the model to deformable particles.

7.2 Outlook

Following the extensive discussion of the LBM-based DNS approach, the six areas below, if addressed, could further improve the applicability of the model to real-world problems:

Shape-dependent hindered settling correlations Correlations for hindered settling, which relate the average bulk settling velocity to the particle volume fraction, the Reynolds number, and shape parameters, are important for practical applications involving suspensions. As discussed in Chapter 1, while these correlations are well established for spherical particles, there is a significant gap in the literature for non-spherical particles. The DNS approach presented in this thesis offers a promising tool for obtaining the necessary data to formulate shape-dependent hindered settling correlations. Future studies should focus on exploring a range of independent shape parameters, such as elongation, roundness, and Hofmann shape entropy, in order to comprehensively characterize the effects of the particle geometry [68]. These correlations could significantly benefit various industrial applications, including separation processes such as thickeners and fluidized bed systems used in chemical reactors. However, it is imperative to complement these computational studies with experimental investigations to validate the results and maintain scientific rigor, highlighting the need for collaborative efforts in this field. Addressing the following areas would further improve the quality of such correlations by allowing for a wider range of parameters.

Utilizing GPUs To obtain statistically meaningful results for shape-dependent hindered settling, especially when considering various independent shape parameters and clustering effects, it becomes necessary to simulate systems with not only thousands, but up to 100 000 particles. This massive increase in computational demand requires a significant leap in computational efficiency. Leveraging modern GPUs is a promising solution, potentially offering more than a tenfold improvement in computational speed. However, harnessing this power requires a major overhaul of the existing particle framework, including the contact model and geometry representation, to ensure CUDA compatibility. This requires a fundamental overhaul of the largely object-oriented data layout, the implementation of generic data structures for seamless synchronization with GPU memory, and the templating of operators to allow for device-side optimization. The latter includes the use of common subexpression elimination (CSE) techniques and subsequent compiler optimizations. Furthermore, the (de)serialization logic must be adapted for device-side execution to facilitate MPI-based communication in multi-GPU setups. These modifications, while challenging, are critical to pushing the boundaries of particle simulation capabilities and enabling the exploration of complex, large-scale particle systems with unprecedented detail and efficiency.

Consideration of turbulence The consideration of high velocities and turbulence in particle-laden flows represents a great potential to extend the scope of possible investigations. For example, it would allow a much wider range of Archimedes numbers to be considered to obtain data for shape-dependent hindered settling correlations. Therefore, such an extension would be very valuable for the development of more comprehensive and widely applicable correlations. Large eddy simulation (LES) offers a promising approach to incorporate turbulence effects, leveraging its existing implementation and validation in OpenLB for single-phase flows [94, 186, 187]. However, HLBM's modified collision step in the lattice Boltzmann equation (LBE) (cf. Chapter 6), which accounts for particle-fluid interactions, requires careful adaptation of the turbulence model. A first validation of the developed model could be performed by considering the fluidization of spherical particles, which is very similar to the hindered settling studies performed in this work, and comparing the results with findings from literature.

Extension of the discrete contact model to concave shapes Extending the discrete contact model presented in Chapter 3 to concave shapes would greatly expand the range of shape parameters that can be investigated. While the force calculations of the current model are theoretically suitable for concave geometries, the contact detection is the limiting factor. Currently, the detection only identifies a single contact region between two solid objects, regardless of the potential for multiple, unconnected contact regions that can occur with concave particles. To overcome this limitation and allow simulation of a wider range of shape parameters, an additional independent step should be introduced after contact detection. This could be achieved by using the breadth-first search (BFS) algorithm to identify the connected components within the bounding box, and then separating these components into distinct, connected regions.

Introduction of a lubrication force correction model HLBM currently overestimates the lubrication force due to its assumption of an incompressible fluid, necessitating the introduction of a lubrication force correction model. To compensate for this overestimation, the particle size is currently virtually increased during contact treatment, resulting in the influence of contact forces even when the particles are not physically touching. This method is likely to introduce errors, particularly at high particle volume fractions above 30%, as observed in Chapter 5. A more refined approach is needed to address these issues, to allow consideration of particle volume fractions beyond the current limit of about 30%, and to improve the overall physical accuracy of the simulations. Instead of relying on virtual size increases, a correction to the HLBM-calculated lubrication force could be implemented by imposing an upper limit. This limit could be set to $\Delta u \rho_\mathrm{f} / \Delta t$, using the local velocity difference Δu, the fluid density

ρ_{f}, and the time step size Δt. The limiting force would therefore be similar to the local back coupling force using the exact difference method (EDM). Such a correction would provide a more physically realistic representation of particle interactions in close proximity without artificially inducing contact forces. To validate this proposed model, a systematic approach could be taken, starting with the rebound of a single sphere, similar to the study performed in Chapter 3. Subsequently, the performance of the model could be also evaluated in hindered settling setups, analogous to those examined in Chapter 5. This stepwise validation process would ensure the robustness and accuracy of the correction model across different flow regimes and particle concentrations, potentially resolving the discrepancies observed at high volume fractions and improving overall accuracy.

Consideration of deformable particles The incorporation of deformable particles into the simulation framework represents a significant opportunity to improve the applicability and accuracy of particle-fluid interaction models. This extension would be particularly valuable for simulating soft materials, biological cells, or certain industrial particles that can change shape in response to fluid forces, particle-particle interactions, and particle-wall interactions. Such a capability would allow the study of complex phenomena such as the squeezing of soft particles in narrow channels or the deformation of particles at high shear rates. These advances could have a significant impact on fields ranging from biomedical applications to advanced materials processing. To achieve this, existing LBM-based approaches that solve nonlinear, transient, large displacement solid mechanics problems [184, 185] could be incorporated into the current framework. The challenge lies in integrating these solid mechanics solvers with HLBM, which uses a confined permeability mapped onto the lattice to account for particle-fluid interactions. A promising solution is to implement a fully coupled dual lattice system: One for fluid dynamics using HLBM, and another dedicated to particle deformation. In this system, the particle lattice would dynamically update the confined permeability mapping in the fluid lattice to reflect changes in particle shape. At the same time, the particle grid would incorporate the forces from the fluid calculations to solve for particle deformation, creating a bidirectional coupling. This integrated approach would retain the efficient particle-fluid interaction framework of HLBM while enabling the simulation of deformable particles, significantly expanding the range of particle systems that can be accurately modeled.

Acknowledgment

This IGF Project of the FEI is/was supported within the programme for promoting the Industrial Collective Research (IGF) of the Federal Ministry of Economic Affairs and Climate Action (BMWK), based on a resolution of the German Parliament. Project 21096 N.

This research was funded by the DFG (German Research Foundation) in the priority program 2045 "Highly specific and multidimensional fractionation of fine particle systems with technical relevance" with grant number KR4259/8-2.

This work was performed on the following computational resources:

- ForHLR II at Karlsruhe Institute of Technology, Germany funded by the Ministry of Science, Research and the Arts Baden-Württemberg and “Deutsche Forschungsgemeinschaft” (DFG)
- BwUniCluster 2.0 provided by the state of Baden-Württemberg through bwHPC
- HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research

Bibliography

[1] Anastasio, C. and Martin, S. T. “Atmospheric Nanoparticles.” In: *Reviews in Mineralogy and Geochemistry* 44.1 (2001), pp. 293–349. DOI: 10.2138/rmg.2001.44.08.

[2] Cheng, T., Gu, X., Yu, T., and Tian, G. “The reflection and polarization properties of non-spherical aerosol particles.” In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 111.6 (2010), pp. 895–906. DOI: 10.1016/j.jqsrt.2009.11.019.

[3] Dubovik, O., Holben, B. N., Lapyonok, T., Sinyuk, A., Mishchenko, M. I., Yang, P., and Slutsker, I. “Non-spherical aerosol retrieval method employing light scattering by spheroids.” In: *Geophysical Research Letters* 29.10 (2002), pp. 54-1-54–4. DOI: https://doi.org/10.1029/2001GL014506.

[4] Huang, Y., Kok, J. F., Saito, M., and Muñoz, O. “Single-scattering properties of ellipsoidal dust aerosols constrained by measured dust shape distributions.” In: *Atmospheric Chemistry and Physics* 23.4 (2023), pp. 2557–2577. DOI: 10.5194/acp-23-2557-2023.

[5] Kómar, L., Wallner, S., and Kocifaj, M. “The significant impact of shape deviations of atmospheric aerosols on light monitoring networks.” In: *Monthly Notices of the Royal Astronomical Society* 512.2 (2022), pp. 1805–1813. DOI: 10.1093/mnras/stac548.

[6] Kumar, P., Robins, A., Vardoulakis, S., and Britter, R. “A review of the characteristics of nanoparticles in the urban atmosphere and the prospects for developing regulatory controls.” In: *Atmospheric Environment* 44.39 (2010), pp. 5035–5052. DOI: 10.1016/j.atmosenv.2010.08.016.

[7] Hargrave, B. "Particle sedimentation in the ocean." In: *Ecological Modelling* 30.3–4 (1985), pp. 229–246. DOI: 10.1016/0304-3800(85)90069-9.

[8] Kolla, V., Henderson, L., and Biscaye, P. E. "Clay mineralogy and sedimentation in the western Indian ocean." In: *Deep Sea Research and Oceanographic Abstracts* 23.10 (1976), pp. 949–961. DOI: 10.1016/0011-7471(76)90825-1.

[9] Rhodes, M., ed. *Introduction to Particle Technology*. 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, 2008. DOI: 10.1002/9780470727102.

[10] Shah, D. S., Moravkar, K. K., Jha, D. K., Lonkar, V., Amin, P. D., and Chalikwar, S. S. "A concise summary of powder processing methodologies for flow enhancement." In: *Heliyon* 9.6 (2023), e16498. DOI: 10.1016/j.heliyon.2023.e16498.

[11] Slomkowski, S., Alemán, J. V., Gilbert, R. G., Hess, M., Horie, K., Jones, R. G., Kubisa, P., Meisel, I., Mormann, W., Penczek, S., and Stepto, R. F. T. "Terminology of polymers and polymerization processes in dispersed systems (IUPAC Recommendations 2011)." In: *Pure and Applied Chemistry* 83.12 (2011), pp. 2229–2259. DOI: 10.1351/pac-rec-10-06-03.

[12] Everett, D. H. "Manual of Symbols and Terminology for Physicochemical Quantities and Units, Appendix II: Definitions, Terminology and Symbols in Colloid and Surface Chemistry." In: *Pure and Applied Chemistry* 31.4 (1972), pp. 577–638. DOI: 10.1351/pac197231040577.

[13] Tanner, R. I. "Review Article: Aspects of non-colloidal suspension rheology." In: *Physics of Fluids* 30.10 (2018). DOI: 10.1063/1.5047535.

[14] Mehdipour, I. and Khayat, K. H. "Understanding the role of particle packing characteristics in rheo-physical properties of cementitious suspensions: A literature review." In: *Construction and Building Materials* 161 (2018), pp. 340–353. DOI: 10.1016/j.conbuildmat.2017.11.147.

[15] Curry, J. A., Ismay, M. J., and Jameson, G. J. "Mine operating costs and the potential impacts of energy and grinding." In: *Minerals Engineering* 56 (2014), pp. 70–80. DOI: 10.1016/j.mineng.2013.10.020.

[16] Schweitzer, P. A., ed. *Handbook of separation techniques for chemical engineers*. 3rd ed. New York, NY, USA: McGraw-Hill, 1997.

[17] Chakrabandhu, K. and Singh, R. K. "Rheological properties of coarse food suspensions in tube flow at high temperatures." In: *Journal of Food Engineering* 66.1 (2005), pp. 117–128. DOI: 10.1016/j.jfoodeng.2004.02.039.

[18] Wasan, D., Nikolov, A., and Aimetti, F. "Texture and stability of emulsions and suspensions: role of oscillatory structural forces." In: *Advances in Colloid and Interface Science* 108–109 (2004), pp. 187–195. DOI: 10.1016/j.cis.2003.10.021.

[19] Chinesta, F. and Ausias, G., eds. *Rheology of Non-spherical Particle Suspensions*. London, UK and Oxford, UK: ISTE Press and Elsevier, 2015. DOI: 10.1016/c2015-0-01208-4.

[20] Chong, Y., Ratkowsky, D., and Epstein, N. "Effect of particle shape on hindered settling in creeping flow." In: *Powder Technology* 23.1 (1979), pp. 55–66. DOI: 10.1016/0032-5910(79)85025-1.

[21] Turney, M. A., Cheung, M. K., Powell, R. L., and McCarthy, M. J. "Hindered settling of rod-like particles measured with magnetic resonance imaging." In: *AIChE Journal* 41.2 (1995), pp. 251–257. DOI: 10.1002/aic.690410207.

[22] Tomkins, M. R., Baldock, T. E., and Nielsen, P. "Hindered settling of sand grains." In: *Sedimentology* 52.6 (2005), pp. 1425–1432. DOI: 10.1111/j.1365-3091.2005.00750.x.

[23] Jirout, T. and Jiroutová, D. "Hindered Settling of Fiber Particles in Viscous Fluids." In: *Processes* 10.99 (2022), p. 1701. DOI: 10.3390/pr10091701.

[24] Mahmoudian, M., Goharpey, F., Behzadnasab, M., and Daneshfar, Z. "Shear thickening and hysteresis in dense suspensions: The effect of particle shape." In: *Journal of Rheology* 68.3 (2024), pp. 479–490. DOI: 10.1122/8.0000799.

[25] Boskovic, L., Altman, I. S., Agranovski, I. E., Braddock, R. D., Myojo, T., and Choi, M. "Influence of particle shape on filtration processes." In: *Aerosol Science and Technology* 39.12 (2005), pp. 1184–1190.

[26] Bagheri, G., Bonadonna, C., Manzella, I., and Vonlanthen, P. "On the characterization of size and shape of irregular particles." In: *Powder Technology* 270 (2015), pp. 141–153. DOI: 10.1016/j.powtec.2014.10.015.

[27] Marquardt, J. E., Arlt, C.-R., Trunk, R., Franzreb, M., and Krause, M. J. "Numerical and experimental examination of the retention of magnetic nanoparticles in magnetic chromatography." In: *Computers & Mathematics with Applications* 89 (2021), pp. 34–43. DOI: 10.1016/j.camwa.2021.02.010.

[28] Vasconcelos, T. F. de, Sapoval, B., Andrade, J. S., Grotberg, J. B., Hu, Y., and Filoche, M. “Particle capture into the lung made simple?” In: *Journal of Applied Physiology* 110.6 (2011), pp. 1664–1673. DOI: 10.1152/japplphysiol.00866.2010.

[29] Capecelatro, J. and Desjardins, O. “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.

[30] Andersson, B., Andersson, R., Håkansson, L., Mortensen, M., Sudiyo, R., and Van Wachem, B. *Computational Fluid Dynamics for Engineers*. Cambridge, UK: Cambridge University Press, 2011. DOI: 10.1017/CBO9781139093590.

[31] Ness, C., Seto, R., and Mari, R. “The Physics of Dense Suspensions.” In: *Annual Review of Condensed Matter Physics* 13.1 (2022), pp. 97–117. DOI: 10.1146/annurev-conmatphys-031620-105938.

[32] Zhu, H. P., Zhou, Z. Y., Yang, R., and Yu, A. “Discrete particle simulation of particulate systems: theoretical developments.” In: *Chemical Engineering Science* 62.13 (2007), pp. 3378–3396. DOI: 10.1016/j.ces.2006.12.089.

[33] Zhu, H. P., Zhou, Z. Y., Yang, R. Y., and Yu, A. B. “Discrete particle simulation of particulate systems: A review of major applications and findings.” In: *Chemical Engineering Science* 63.23 (2008), pp. 5728–5770. DOI: 10.1016/j.ces.2008.08.006.

[34] Guises, R., Xiang, J., Latham, J.-P., and Munjiza, A. “Granular packing: numerical simulation and the characterisation of the effect of particle shape.” In: *Granular Matter* 11 (2009), pp. 281–292. DOI: 10.1007/s10035-009-0148-0.

[35] Houlsby, G. “Potential particles: a method for modelling non-circular particles in DEM.” In: *Computers and Geotechnics* 36.6 (2009), pp. 953–959. DOI: 10.1016/j.compgeo.2009.03.001.

[36] Kodam, M., Bharadwaj, R., Curtis, J., Hancock, B., and Wassgren, C. “Cylindrical object contact detection for use in discrete element method simulations, Part II—Experimental validation.” In: *Chemical Engineering Science* 65.22 (2010), pp. 5863–5871. DOI: 10.1016/j.ces.2010.08.007.

[37] Lin, X. and Ng, T.-T. "Contact detection algorithms for three-dimensional ellipsoids in discrete element modelling." In: *International Journal for Numerical and Analytical Methods in Geomechanics* 19.9 (1995), pp. 653–659. DOI: 10.1002/nag.1610190905.

[38] Fraige, F. Y., Langston, P. A., and Chen, G. Z. "Distinct element modelling of cubic particle packing and flow." In: *Powder Technology* 186.3 (2008), pp. 224–240. DOI: 10.1016/j.powtec.2007.12.009.

[39] Matuttis, H., Luding, S., and Herrmann, H. "Discrete element simulations of dense packings and heaps made of spherical and non-spherical particles." In: *Powder technology* 109.1-3 (2000), pp. 278–292. DOI: 10.1016/S0032-5910(99)00243-0.

[40] Nezami, E. G., Hashash, Y. M., Zhao, D., and Ghaboussi, J. "A fast contact detection algorithm for 3-D discrete element method." In: *Computers and geotechnics* 31.7 (2004), pp. 575–587. DOI: 10.1016/j.compgeo.2004.08.002.

[41] Nolan, G. T. and Kavanagh, P. E. "Random packing of nonspherical particles." In: *Powder Technology* 84.3 (1995), pp. 199–205. DOI: 10.1016/0032-5910(95)98237-S.

[42] Kruggel-Emden, H., Rickelt, S., Wirtz, S., and Scherer, V. "A study on the validity of the multi-sphere Discrete Element Method." In: *Powder Technology* 188.2 (2008), pp. 153–165. DOI: 10.1016/j.powtec.2008.04.037.

[43] Rakotonirina, A. D. and Wachs, A. "Grains3D, a flexible DEM approach for particles of arbitrary convex shape—Part II: Parallel implementation and scalable performance." In: *Powder technology* 324 (2018), pp. 18–35. DOI: 10.1016/j.powtec.2017.10.033.

[44] Rakotonirina, A. D., Delenne, J.-Y., Radjai, F., and Wachs, A. "Grains3D, a flexible DEM approach for particles of arbitrary convex shape—Part III: extension to non-convex particles modelled as glued convex particles." In: *Computational Particle Mechanics* 6.1 (2019), pp. 55–84. DOI: 10.1007/s40571-018-0198-3.

[45] Wachs, A., Girolami, L., Vinay, G., and Ferrer, G. "Grains3D, a flexible DEM approach for particles of arbitrary convex shape—Part I: Numerical model and validations." In: *Powder technology* 224 (2012), pp. 374–389. DOI: 10.1016/j.powtec.2012.03.023.

[46] Qiu, L.-c. and Wu, C.-y. “A hybrid DEM/CFD approach for solid-liquid flows.” In: *Journal of Hydrodynamics* 26 (2014), pp. 19–25. DOI: 10.1016/S1001-6058(14)60003-2.

[47] Sun, R. and Xiao, H. “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.

[48] Weers, M., Hansen, L., Schulz, D., Benker, B., Wollmann, A., Kykal, C., Kruggel-Emden, H., and Weber, A. P. “Development of a Model for the Separation Characteristics of a Deflector Wheel Classifier Including Particle Collision and Rebound Behavior.” In: *Minerals* 12.4 (2022), p. 480. DOI: 10.3390/min12040480.

[49] Fu, J., Chen, S., Chen, P., and Wen, C. “Particle-resolved simulation on viscous flow past random and ordered arrays of hot ellipsoidal particles.” In: *International Journal of Multiphase Flow* 142 (2021), p. 103736. DOI: 10.1016/j.ijmultiphaseflow.2021.103736.

[50] Rosemann, T., Reinecke, S. R., and Kruggel-Emden, H. “Analysis of Mobility Effects in Particle-Gas Flows by Particle-Resolved LBM-DEM Simulations.” In: *Chemie Ingenieur Technik* 93.1–2 (2020), pp. 223–236. DOI: 10.1002/cite.202000204.

[51] Uhlmann, M. and Doychev, T. “Sedimentation of a dilute suspension of rigid spheres at intermediate Galileo numbers: the effect of clustering upon the particle motion.” In: *Journal of Fluid Mechanics* 752 (2014), pp. 310–348. DOI: 10.1017/jfm.2014.330.

[52] Walayat, K., Zhang, Z., Usman, K., Chang, J., and Liu, M. “Fully resolved simulations of thermal convective suspensions of elliptic particles using a multigrid fictitious boundary method.” In: *International Journal of Heat and Mass Transfer* 139 (2019), pp. 802–821. DOI: 10.1016/j.ijheatmasstransfer.2019.05.068.

[53] Willen, D. P., Sierakowski, A. J., Zhou, G., and Prosperetti, A. “Continuity waves in resolved-particle simulations of fluidized beds.” In: *Physical Review Fluids* 2.11 (2017), p. 114305. DOI: 10.1103/PhysRevFluids.2.114305.

[54] Willen, D. P. and Prosperetti, A. “Resolved simulations of sedimenting suspensions of spheres.” In: *Physical Review Fluids* 4.1 (2019), p. 014304. DOI: 10.1103/PhysRevFluids.4.014304.

[55] Yao, Y., Criddle, C. S., and Fringer, O. B. "The effects of particle clustering on hindered settling in high-concentration particle suspensions." In: *Journal of Fluid Mechanics* 920 (2021), A40. DOI: 10.1017/jfm.2021.470.

[56] Zaidi, A. A., Tsuji, T., and Tanaka, T. "Hindered Settling Velocity & Structure Formation during Particle Settling by Direct Numerical Simulation." In: *Procedia Engineering* 102 (2015), pp. 1656–1666. DOI: 10.1016/j.proeng.2015.01.302.

[57] Peskin, C. S. "The immersed boundary method." In: *Acta numerica* 11 (2002), pp. 479–517. DOI: 10.1017/S0962492902000077.

[58] Uhlmann, M. "An immersed boundary method with direct forcing for the simulation of particulate flows." In: *Journal of Computational Physics* 209.2 (2005), pp. 448–476. DOI: 10.1016/j.jcp.2005.03.017.

[59] Nagata, T., Hosaka, M., Takahashi, S., Shimizu, K., Fukuda, K., and Obayashi, S. "A simple collision algorithm for arbitrarily shaped objects in particle-resolved flow simulation using an immersed boundary method." In: *International Journal for Numerical Methods in Fluids* 92.10 (2020), pp. 1256–1273. DOI: 10.1002/fld.4826.

[60] Boon, C. W., Houlsby, G., and Utili, S. "A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method." In: *Computers and Geotechnics* 44 (2012), pp. 73–82. DOI: 10.1016/j.compgeo.2012.03.012.

[61] Noble, D. R. and Torczynski, J. R. "A Lattice-Boltzmann Method for Partially Saturated Computational Cells." In: *International Journal of Modern Physics C* 09.08 (1998), pp. 1189–1201. DOI: 10.1142/S0129183198001084.

[62] Haussmann, M., Hafen, N., Raichle, F., Trunk, R., Nirschl, H., and Krause, M. J. "Galilean invariance study on different lattice Boltzmann fluid–solid interface approaches for vortex-induced vibrations." In: *Computers & Mathematics with Applications* 80 (2020), pp. 671–691. DOI: 10.1016/j.camwa.2020.04.022.

[63] Rettinger, C. and Rüde, U. "A comparative study of fluid-particle coupling methods for fully resolved lattice Boltzmann simulations." In: *Computers & Fluids* 154 (2017), pp. 74–89. DOI: 10.1016/j.compfluid.2017.05.033.

[64] Li, X., Wang, F., Zhang, D., Gu, S., and Gao, X. “Fluid-solid interaction simulation for particles and walls of arbitrary polygonal shapes with a coupled LBM-IMB-DEM method.” In: *Powder Technology* 356 (2019), pp. 177–192. DOI: 10.1016/j.powtec.2019.08.006.

[65] Succi, S. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. New York, NY, USA: Oxford University Press, 2001. DOI: 10.1093/oso/9780198503989.001.0001.

[66] Krause, M. J., Klemens, F., Henn, T., Trunk, R., and Nirschl, H. “Particle flow simulations with homogenised lattice Boltzmann methods.” In: *Particuology* 34 (2017), pp. 1–13. DOI: 10.1016/j.partic.2016.11.001.

[67] Trunk, R., Marquardt, J., Thäter, G., Nirschl, H., and Krause, M. J. “Towards the simulation of arbitrarily shaped 3D particles using a homogenised lattice Boltzmann method.” In: *Computers & Fluids* 172 (2018), pp. 621–631. DOI: 10.1016/j.compfluid.2018.02.027.

[68] Trunk, R., Bretl, C., Thäter, G., Nirschl, H., Dorn, M., and Krause, M. J. “A Study on Shape-Dependent Settling of Single Particles with Equal Volume Using Surface Resolved Simulations.” In: *Computation* 9.44 (2021), p. 40. DOI: 10.3390/computation9040040.

[69] Trunk, R., Weckerle, T., Hafen, N., Thäter, G., Nirschl, H., and Krause, M. J. “Revisiting the Homogenized Lattice Boltzmann Method with Applications on Particulate Flows.” In: *Computation* 9.22 (2021), p. 11. DOI: 10.3390/computation9020011.

[70] Trunk, R. A. “Numerical Investigation of the Settling Behavior of Non-Spherical Particles - Application of Homogenized Lattice Boltzmann Methods.” PhD thesis. Karlsruhe, Germany: Karlsruhe Institute of Technologie (KIT), 2021. DOI: 10.5445/IR/1000136875.

[71] Marquardt, J. E., Römer, U. J., Nirschl, H., and Krause, M. J. “A discrete contact model for complex arbitrary-shaped convex geometries.” In: *Particuology* 80 (2023), pp. 180–191. DOI: 10.1016/j.partic.2022.12.005.

[72] Marquardt, J. E., Hafen, N., and Krause, M. J. “A novel particle decomposition scheme to improve parallel performance of fully resolved particulate flow simulations.” In: *Journal of Computational Science* 78 (2024), p. 102263. DOI: 10.1016/j.jocs.2024.102263.

[73] Marquardt, J. E., Hafen, N., and Krause, M. J. "A novel model for direct numerical simulation of suspension dynamics with arbitrarily shaped convex particles." In: *Computer Physics Communications* 304 (2024), p. 109321. DOI: 10.1016/j.cpc.2024.109321.

[74] Marquardt, J. E. and Krause, M. J. "A Review of the Homogenized Lattice Boltzmann Method for Particulate Flow Simulations: From Fundamentals to Applications." In: *Powders* 3.4 (2024), pp. 500–530. DOI: 10.3390/powders3040027.

[75] Kummerländer, A., Bingert, T., Bukreev, F., Czelusniak, L. E., Dapelo, D., Hafen, N., Heinzelmann, M., Ito, S., Jeßbberger, J., Kusumaatmaja, H., Marquardt, J. E., Rennick, M., Pertzel, T., Prinz, F., Sadric, M., Schecher, M., Simonis, S., Sitter, P., Teutscher, D., Zhong, M., and Krause, M. J. *OpenLB Release 1.7: Open Source Lattice Boltzmann Code*. Version 1.7.0. 2024. DOI: 10.5281/zenodo.10684609.

[76] Steinour, H. H. "Rate of sedimentation: Nonflocculated Suspensions of Uniform Spheres." In: *Industrial & Engineering Chemistry* 36.7 (1944), pp. 618–624. DOI: 10.1021/ie50415a005.

[77] Richardson, J. F. and Zaki, W. N. "Sedimentation and fluidisation: Part I." In: *Chemical Engineering Research and Design* 75 (1954), S82–S100. DOI: 10.1016/S0263-8762(97)80006-8.

[78] Barnea, E. and Mizrahi, J. "A generalized approach to the fluid dynamics of particulate systems." In: *The Chemical Engineering Journal* 5.2 (1973), pp. 171–189. DOI: 10.1016/0300-9467(73)80008-5.

[79] Garside, J. and Al-Dibouni, M. R. "Velocity-voidage relationships for fluidization and sedimentation in solid-liquid systems." In: *Industrial & engineering chemistry process design and development* 16.2 (1977). Citation Key: garside1977velocity, pp. 206–214.

[80] Di Felice, R. "Hydrodynamics of liquid fluidisation." In: *Chemical Engineering Science* 50.8 (1995), pp. 1213–1245. DOI: 10.1016/0009-2509(95)98838-6.

[81] Di Felice, R. "The sedimentation velocity of dilute suspensions of nearly monosized spheres." In: *International Journal of Multiphase Flow* 25.4 (1999), pp. 559–574. DOI: 10.1016/S0301-9322(98)00084-6.

[82] Hamid, A., Molina, J. J., and Yamamoto, R. “Direct numerical simulations of sedimenting spherical particles at non-zero Reynolds number.” In: *RSC Advances* 4.96 (2014), pp. 53681–53693. DOI: 10.1039/C4RA11025K.

[83] Amiri Delouei, A., Karimnejad, S., and He, F. “Direct Numerical Simulation of pulsating flow effect on the distribution of non-circular particles with increased levels of complexity: IB-LBM.” In: *Computers & Mathematics with Applications* 121 (2022), pp. 115–130. DOI: 10.1016/j.camwa.2022.07.005.

[84] Karimnejad, S., Amiri Delouei, A., Nazari, M., Shahmardan, M., and Mohamad, A. “Sedimentation of elliptical particles using Immersed Boundary – Lattice Boltzmann Method: A complementary repulsive force model.” In: *Journal of Molecular Liquids* 262 (2018), pp. 180–193. DOI: 10.1016/j.molliq.2018.04.075.

[85] Hu, Y., Li, D., Shu, S., and Niu, X. “Modified momentum exchange method for fluid-particle interactions in the lattice Boltzmann method.” In: *Physical Review E* 91.3 (2015), p. 033301. DOI: 10.1103/physreve.91.033301.

[86] Hafen, N., Dittler, A., and Krause, M. J. “Simulation of particulate matter structure detachment from surfaces of wall-flow filters applying lattice Boltzmann methods.” In: *Computers & Fluids* 239 (2022), p. 105381. DOI: 10.1016/j.compfluid.2022.105381.

[87] Hafen, N., Marquardt, J. E., Dittler, A., and Krause, M. J. “Simulation of Particulate Matter Structure Detachment from Surfaces of Wall-Flow Filters for Elevated Velocities Applying Lattice Boltzmann Methods.” In: *Fluids* 8.3 (2023), p. 99. DOI: 10.3390/fluids8030099.

[88] Hafen, N., Marquardt, J. E., Dittler, A., and Krause, M. J. “Simulation of Dynamic Rearrangement Events in Wall-Flow Filters Applying Lattice Boltzmann Methods.” In: *Fluids* 8.7 (2023), p. 213. DOI: 10.3390/fluids8070213.

[89] Karimnejad, S., Delouei, A. A., Başağaoğlu, H., Nazari, M., Shahmardan, M., Falcucci, G., Lauricella, M., and Succi, S. “A Review on Contact and Collision Methods for Multi-Body Hydrodynamic Problems in Complex Flows.” In: *Communications in Computational Physics* 32.4 (2022), pp. 899–950. DOI: 10.4208/cicp.re-2022-0041.

[90] Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., and Viggen, E. M. *The Lattice Boltzmann Method. Principles and practice*. Cham, Switzerland: Springer International Publishing, 2017. DOI: 10.1007/978-3-319-44649-3.

[91] Sukop, M. C. and Thorne, D. T. *Lattice Boltzmann Modeling. An Introduction for Geoscientists and Engineers*. 2nd ed., corrected printing. Berlin/Heidelberg, Germany: Springer, 2006. DOI: 10.1007/978-3-540-27982-2.

[92] Bhatnagar, P. L., Gross, E. P., and Krook, M. "A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems." In: *Physical Review* 94.3 (1954), pp. 511–525. DOI: 10.1103/PhysRev.94.511.

[93] Qian, Y. H., D'Humières, D., and Lallemand, P. "Lattice BGK Models for Navier-Stokes Equation." In: *Europhysics Letters (EPL)* 17.6 (1992), pp. 479–484. DOI: 10.1209/0295-5075/17/6/001.

[94] Krause, M. J., Kummerländer, A., Avis, S. J., Kusumaatmaja, H., Dapelo, D., Klemens, F., Gaedtke, M., Hafen, N., Mink, A., Trunk, R., Marquardt, J. E., Maier, M.-L., Haussmann, M., and Simonis, S. "OpenLB—Open source lattice Boltzmann code." In: *Computers & Mathematics with Applications* 81 (2021), pp. 258–288. DOI: 10.1016/j.camwa.2020.04.033.

[95] Kupershtokh, A., Medvedev, D., and Karpov, D. "On Equations of State in a Lattice Boltzmann Method." In: *Computers & Mathematics with Applications* 58 (2009), pp. 965–974. DOI: 10.1016/j.camwa.2009.02.024.

[96] Wen, B., Zhang, C., Tu, Y., Wang, C., and Fang, H. "Galilean invariant fluid–solid interfacial dynamics in lattice Boltzmann simulations." In: *Journal of Computational Physics* 266 (2014), pp. 161–170. DOI: 10.1016/j.jcp.2014.02.018.

[97] Henn, T., Thäter, G., Dörfler, W., Nirschl, H., and Krause, M. J. "Parallel dilute particulate flow simulations in the human nasal cavity." In: *Computers & Fluids* 124 (2016), pp. 197–207. DOI: 10.1016/j.compfluid.2015.08.002.

[98] Yin, X. and Koch, D. L. "Hindered settling velocity and microstructure in suspensions of solid spheres with moderate Reynolds numbers." In: *Physics of Fluids* 19.9 (2007), p. 093302. DOI: 10.1063/1.2764109.

[99] Schiller, L. and Neumann, A. "Über die grundlegenden Berechnungen bei der Schwerkraftaufbereitung." In: *Z. Vereines Deutscher Ingenieure* 77 (1933), pp. 318–321.

[100] Ergun, S. “Fluid flow through packed columns.” In: *Chemical Engineering Progress* 48.2 (1952), pp. 89–94.

[101] Ginzbourg, I. and d’Humières, D. “Local second-order boundary methods for lattice Boltzmann models.” In: *Journal of Statistical Physics* 84.5–6 (1996), pp. 927–971. DOI: 10.1007/bf02174124.

[102] Krause, M., Avis, S., Kusumaatmaja, H., Dapelo, D., Gaedtke, M., Hafen, N., Haußmann, M., Jeppener-Haltenhoff, J., Kronberg, L., Kummerländer, A., Marquardt, J. E., Pertzel, T., Simonis, S., Trunk, R., Wu, M., and Zarth, A. *OpenLB Release 1.4: Open Source Lattice Boltzmann Code*. Version 1.4. 2020. DOI: 10.5281/zenodo.4279263.

[103] Hart, J. C. “Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces.” In: *The Visual Computer* 12.10 (1996), pp. 527–545. DOI: 10.1007/s003710050084.

[104] Haugo, S., Stahl, A., and Brekke, E. “Continuous Signed Distance Functions for 3D Vision.” In: *2017 International Conference on 3D Vision (3DV)*. 2017, pp. 116–125. DOI: 10.1109/3DV.2017.00023.

[105] Nassauer, B. and Kuna, M. “Contact forces of polyhedral particles in discrete element method.” In: *Granular Matter* 15.3 (2013), pp. 349–355. DOI: 10.1007/s10035-013-0417-9.

[106] Carvalho, A. S. and Martins, J. M. “Exact restitution and generalizations for the Hunt–Crossley contact model.” In: *Mechanism and Machine Theory* 139 (2019), pp. 174–194. DOI: 10.1016/j.mechmachtheory.2019.03.028.

[107] Alves, J., Peixinho, N., da Silva, M. T., Flores, P., and Lankarani, H. M. “A comparative study of the viscoelastic constitutive models for frictionless contact interfaces in solids.” In: *Mechanism and Machine Theory* 85 (2015), pp. 172–188. DOI: 10.1016/j.mechmachtheory.2014.11.020.

[108] Swope, W. C., Andersen, H. C., Berens, P. H., and Wilson, K. R. “A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters.” In: *The Journal of Chemical Physics* 76.1 (1982), pp. 637–649. DOI: 10.1063/1.442716.

[109] Verlet, L. “Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules.” In: *Physical Review* 159.1 (1967), pp. 98–103. DOI: 10.1103/PhysRev.159.98.

[110] Džiugys, A. and Peters, B. “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.

[111] Popov, V. L., Heß, M., and Willert, E. *Handbook of Contact Mechanics. Exact Solutions of Axisymmetric Contact Problems*. Berlin/Heidelberg, Germany: Springer, 2019. DOI: 10.1007/978-3-662-58709-6.

[112] Popov, V. L. *Contact Mechanics and Friction. Physical Principles and Applications*. 2nd ed. Berlin/Heidelberg, Germany: Springer, 2017. DOI: 10.1007/978-3-662-53081-8.

[113] Park, J. “Modeling the dynamics of fabric in a rotating horizontal drum.” PhD thesis. West Lafayette, IN, USA: Purdue University, 2003.

[114] Li, X. “An Experimental and Numerical Study of Normal Particle Collisions in a Viscous Liquid.” PhD thesis. Pasadena, CA, USA: California Institute of Technology, 2010. DOI: 10.7907/S5V0-3E25.

[115] Kummerländer, A., Avis, S., Kusumaatmaja, H., Bukreev, F., Crocoll, M., Dapelo, D., Großmann, S., Hafen, N., Ito, S., Jeßberger, J., Kummer, E., Marquardt, J. E., Mödl, J., Pertzel, T., Prinz, F., Raichle, F., Sadric, M., Schecher, M., Teutscher, D., Simonis, S., and Krause, M. J. *OpenLB User Guide: Associated with Release 1.6 of the Code*. 2023. DOI: 10.48550/arXiv.2307.11752. URL: https://doi.org/10.48550/arXiv.2307.11752.

[116] Martys, N. S. and Mountain, R. D. “Velocity Verlet algorithm for dissipative-particle-dynamics-based models of suspensions.” In: *Physical Review E* 59.3 (1999), pp. 3733–3736. DOI: 10.1103/physreve.59.3733.

[117] Fraige, F. and Langston, P. “Integration schemes and damping algorithms in distinct element models.” In: *Advanced Powder Technology* 15.2 (2004), pp. 227–245. DOI: 10.1163/156855204773644454.

[118] Toxvaerd, S., Heilmann, O. J., and Dyre, J. C. “Energy conservation in molecular dynamics simulations of classical systems.” In: *The Journal of Chemical Physics* 136.22 (2012). DOI: 10.1063/1.4726728.

[119] Hairer, E., Lubich, C., and Wanner, G. “Geometric numerical integration illustrated by the Störmer–Verlet method.” In: *Acta Numerica* 12 (2003), pp. 399–450. DOI: 10.1017/s0962492902000144.

[120] Nirschl, H. *Partikelbewegungen in Scherströmungen ohne und mit Berücksichtigung des Einflusses angrenzender Wände*. Düsseldorf, Germany: VDI Verlag, 1997.

[121] Oliver, D. R. "The sedimentation of suspensions of closely-sized spherical particles." In: *Chemical Engineering Science* 15.3 (1961), pp. 230–242. DOI: 10.1016/0009-2509(61)85026-4.

[122] Kawamoto, R., Andò, E., Viggiani, G., and Andrade, J. E. "Level set discrete element method for three-dimensional computations with triaxial case study." In: *Journal of the Mechanics and Physics of Solids* 91 (2016), pp. 1–13. DOI: 10.1016/j.jmps.2016.02.021.

[123] Haven, D. L. H. van der, Fragkopoulos, I. S., and Elliott, J. A. "A physically consistent Discrete Element Method for arbitrary shapes using Volume-interacting Level Sets." In: *Computer Methods in Applied Mechanics and Engineering* 414 (2023), p. 116165. DOI: 10.1016/j.cma.2023.116165.

[124] Wang, S., Liang, D., and Ji, S. "DEM study on mixing behaviors of concave-shaped particles in rotating drum based on level-set method." In: *Powder Technology* 430 (2023), p. 118961. DOI: 10.1016/j.powtec.2023.118961.

[125] Blais, B., Barbeau, L., Bibeau, V., Gauvin, S., Geitani, T. E., Golshan, S., Kamble, R., Mirakhori, G., and Chaouki, J. "Lethe: An open-source parallel high-order adaptative CFD solver for incompressible flows." In: *SoftwareX* 12 (2020), p. 100579. DOI: 10.1016/j.softx.2020.100579.

[126] El Geitani, T., Golshan, S., and Blais, B. "Toward High-Order CFD-DEM: Development and Validation." In: *Industrial & Engineering Chemistry Research* 62.2 (2023), pp. 1141–1159. DOI: 10.1021/acs.iecr.2c03546.

[127] Golshan, S., Munch, P., Gassmöller, R., Kronbichler, M., and Blais, B. "Lethe-DEM: an open-source parallel discrete element solver with load balancing." In: *Computational Particle Mechanics* 10.1 (2022), pp. 77–96. DOI: 10.1007/s40571-022-00478-6.

[128] Flores, P. "Contact mechanics for dynamical systems: a comprehensive review." In: *Multibody System Dynamics* (2021). DOI: 10.1007/s11044-021-09803-y.

[129] Hafen, N., Thieringer, J. R., Meyer, J., Krause, M. J., and Dittler, A. "Numerical investigation of detachment and transport of particulate structures in wall-flow filters using lattice Boltzmann methods." In: *J. Fluid Mech.* 956 (2023), A30. DOI: 10.1017/jfm.2023.35.

[130] Chen, Y., Jin, G., Zhang, P., Galindo-Torres, S. A., Scheuermann, A., and Li, L. "An efficient framework for particle-fluid interaction using Discrete Element Lattice Boltzmann Method: Coupling scheme and periodic boundary condition." In: *Computers & Fluids* 208 (2020), p. 104613. DOI: 10.1016/j.compfluid.2020.104613.

[131] Tang, H., Song, R., Dong, Y., and Song, X. "Measurement of Restitution and Friction Coefficients for Granular Particles and Discrete Element Simulation for the Tests of Glass Beads." In: *Materials* 12.19 (2019), p. 3170. DOI: 10.3390/ma12193170.

[132] Barths, H., Hasse, C., Bikas, G., and Peters, N. "Simulation of combustion in direct injection diesel engines using a eulerian particle flamelet model." In: *Proceedings of the Combustion Institute* 28.1 (2000), pp. 1161–1168. DOI: 10.1016/S0082-0784(00)80326-4.

[133] Kashiwaya, K., Noumachi, T., Hiroyoshi, N., Ito, M., and Tsunekawa, M. "Effect of particle shape on hydrocyclone classification." In: *Powder Technology* 226 (2012), pp. 147–156. DOI: 10.1016/j.powtec.2012.04.036.

[134] Schwister, K., ed. *Taschenbuch der Verfahrenstechnik*. 5th ed. Munich, Germany: Carl Hanser Verlag, 2017.

[135] Merkus, H. G. "Particle Size, Size Distributions and Shape." In: *Particle Size Measurements: Fundamentals, Practice, Quality*. Dordrecht, Netherlands, 2009, pp. 13–42. DOI: 10.1007/978-1-4020-9016-5_2.

[136] Dal Pont, J.-P., ed. *Process Engineering and Industrial Management*. Hoboken, NJ, USA: John Wiley & Sons, 2013. DOI: 10.1002/9781118562130.

[137] Johansson, R. and Evertsson, M. "CFD simulation of a centrifugal air classifier used in the aggregate industry." In: *Minerals Engineering* 63 (2014), pp. 149–156. DOI: 10.1016/j.mineng.2014.03.013.

[138] Meier, J., Klein, G.-M., and Kottke, V. "Crossflow filtration as a new method of wet classification of ultrafine particles." In: *Separation and Purification Technology* 26.1 (2002), pp. 43–50. DOI: 10.1016/S1383-5866(01)00115-0.

[139] Saunders, S. R. and Roberts, C. B. "Size-selective fractionation of nanoparticles at an application scale using CO2 gas-expanded liquids." In: *Nanotechnology* 20.47 (2009), p. 475605. DOI: 10.1088/0957-4484/20/47/475605.

[140] Saunders, S. R. and Roberts, C. B. “Nanoparticle separation and deposition processing using gas expanded liquid technology.” In: *Current Opinion in Chemical Engineering* 1.2 (2012), pp. 91–101. DOI: 10.1016/j.coche.2011.12.004.

[141] Peukert, W. and Wadenpohl, C. “Industrial separation of fine particles with difficult dust properties.” In: *Powder technology* 118.1-2 (2001), pp. 136–148. DOI: 10.1016/S0032-5910(01)00304-7.

[142] Williams, A., Varela, E., Meehan, E., and Tribe, K. “Characterisation of nanoparticulate systems by hydrodynamic chromatography.” In: *International journal of pharmaceutics* 242.1-2 (2002), pp. 295–299. DOI: 10.1016/S0378-5173(02)00191-6.

[143] Liu, J.-f., Yu, S.-j., Yin, Y.-g., and Chao, J.-b. “Methods for separation, identification, characterization and quantification of silver nanoparticles.” In: *TrAC Trends in Analytical Chemistry* 33 (2012), pp. 95–106. DOI: 10.1016/j.trac.2011.10.010.

[144] Spelter, L. E., Meyer, K., and Nirschl, H. “Screening of colloids by semicontinuous centrifugation.” In: *Chemical engineering & technology* 35.8 (2012), pp. 1486–1494. DOI: 10.1002/ceat.201200050.

[145] Kemenade, E. van, Brouwers, B., and Benthum, R. van. “Centrifugal separation with emphasis on the rotational particle separator.” In: *ChemBioEng Reviews* 1.6 (2014), pp. 262–272. DOI: 10.1002/cben.201400027.

[146] Lecrivain, G., Rayan, R., Hurtado, A., and Hampel, U. “Using quasi-DNS to investigate the deposition of elongated aerosol particles in a wavy channel flow.” In: *Computers & Fluids* 124 (2016), pp. 78–85. DOI: 10.1016/j.compfluid.2015.10.012.

[147] Krause, M. J. “Fluid Flow Control and Simulation in Process Engineering: Homogenised Lattice Boltzmann Methods.” Habilitation thesis. Karlsruhe, Germany: Karlsruhe Institute of Technology (KIT), 2022.

[148] Jones, J. E. “On the determination of molecular fields.—II. From the equation of state of a gas.” In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 106.738 (1924), pp. 463–477. DOI: 10.1098/rspa.1924.0082.

[149] Verwey, E. J. W. “Theory of the stability of lyophobic colloids.” In: *The Journal of Physical Chemistry* 51.3 (1947), pp. 631–636. DOI: 10.1021/j150453a001.

[150] Cundall, P. A. and Strack, O. D. “A discrete numerical model for granular assemblies.” In: *geotechnique* 29.1 (1979), pp. 47–65. DOI: 10.1680/geot.1979.29.1.47.

[151] Yow, H., Pitt, M., and Salman, A. “Drag correlations for particles of regular shape.” In: *Advanced Powder Technology* 16.4 (2005), pp. 363–372. DOI: 10.1163/1568552054194221.

[152] Blais, B., Tucny, J.-M., Vidal, D., and Bertrand, F. “A conservative lattice Boltzmann model for the volume-averaged Navier–Stokes equations based on a novel collision operator.” In: *Journal of Computational Physics* 294 (2015), pp. 258–273. DOI: 10.1016/j.jcp.2015.03.036.

[153] Munjiza, A. and Andrews, K. “Penalty function method for combined finite–discrete element systems comprising large number of separate bodies.” In: *International Journal for numerical methods in engineering* 49.11 (2000), pp. 1377–1396. DOI: 10.1002/1097-0207(20001220)49:11<1377::AID-NME6>3.0.CO;2-B.

[154] Zhang, H., Trias, F. X., Oliva, A., Yang, D., Tan, Y., Shu, S., and Sheng, Y. “PIBM: Particulate immersed boundary method for fluid–particle interaction problems.” In: *Powder technology* 272 (2015), pp. 1–13. DOI: 10.1016/j.powtec.2014.11.025.

[155] Maier, M.-L., Milles, S., Schuhmann, S., Guthausen, G., Nirschl, H., and Krause, M. “Fluid flow simulations verified by measurements to investigate adsorption processes in a static mixer.” In: *Computers & Mathematics with Applications* 76.11 (2018), pp. 2744–2757. DOI: 10.1016/j.camwa.2018.08.066.

[156] Bukreev, F., Simonis, S., Kummerländer, A., Jeßberger, J., and Krause, M. J. “Consistent lattice Boltzmann methods for the volume averaged Navier–Stokes equations.” In: *Journal of Computational Physics* (2023), p. 112301. DOI: 10.1016/j.jcp.2023.112301.

[157] Höcker, S., Trunk, R., Dörfler, W., and Krause, M. “Towards the simulations of inertial dense particulate flows with a volume-averaged lattice Boltzmann method.” In: *Computers & Fluids* 166 (2018), pp. 152–162. DOI: 10.1016/j.compfluid.2018.02.011.

[158] Shan, X. and Chen, H. “Lattice Boltzmann model for simulating flows with multiple phases and components.” In: *Physical review E* 47.3 (1993), p. 1815. DOI: 10.1103/PhysRevE.47.1815.

[159] Trunk, R., Henn, T., Dörfler, W., Nirschl, H., and Krause, M. "Inertial dilute particulate fluid flow simulations with an Euler–Euler lattice Boltzmann method." In: *Journal of Computational Science* 17 (2016), pp. 438–445. DOI: 10.1016/j.jocs.2016.03.013.

[160] Marchisio, D. L. and Fox, R. O. "Solution of population balance equations using the direct quadrature method of moments." In: *Journal of Aerosol Science* 36.1 (2005), pp. 43–73. DOI: 10.1016/j.jaerosci.2004.07.009.

[161] McGraw, R. "Description of aerosol dynamics by the quadrature method of moments." In: *Aerosol science and Technology* 27.2 (1997), pp. 255–265. DOI: 10.1080/02786829708965471.

[162] Łukaszewicz, G. and Kalita, P. *Navier–Stokes Equations. An Introduction with Applications*. Cham, Switzerland: Springer International Publishing, 2016. DOI: 10.1007/978-3-319-27760-8.

[163] Reddy, J. N. *An Introduction to Continuum Mechanics, Second Edition*. 2nd ed. Cambridge, UK: Cambridge University Press, 2013. DOI: 10.1017/CBO9781139178952.

[164] Spaid, M. A. A. and Phelan, F. R. "Lattice Boltzmann methods for modeling microscale flow in fibrous porous media." In: *Physics of Fluids* 9.9 (1997), pp. 2468–2474. DOI: 10.1063/1.869392.

[165] Simonis, S., Hafen, N., Jeßberger, J., Dapelo, D., Thäter, G., and Krause, M. J. *Homogenized lattice Boltzmann methods for fluid flow through porous media – part I: kinetic model derivation*. 2023. DOI: 10.48550/arXiv.2310.14746.

[166] Guo, Z. and Zhao, T. S. "Lattice Boltzmann model for incompressible flows through porous media." In: *Phys. Rev. E* 66.3 (2002). Publisher: American Physical Society, p. 036304. DOI: 10.1103/PhysRevE.66.036304.

[167] Ditscherlein, R., Furat, O., Löwer, E., Mehnert, R., Trunk, R., Leißner, T., Krause, M. J., Schmidt, V., and Peuker, U. A. "PARROT: A Pilot Study on the Open Access Provision of Particle-Discrete Tomographic Datasets." In: *Microscopy and Microanalysis* 28.2 (2022), pp. 350–360. DOI: 10.1017/s143192762101391x.

[168] Kummerländer, A., Avis, S., Kusumaatmaja, H., Bukreev, F., Crocoll, M., Dapelo, D., Hafen, N., Ito, S., Jeßberger, J., Marquardt, J. E., Mödl, J., Pertzel, T., Prinz, F., Raichle, F., Schecher, M., Simonis, S., Teutscher, D.,

and Krause, M. J. *OpenLB Release 1.6: Open Source Lattice Boltzmann Code*. Version 1.6. 2023. DOI: 10.5281/zenodo.7773497.

[169] Dapelo, D., Trunk, R., Krause, M. J., and Bridgeman, J. "Towards Lattice-Boltzmann modelling of unconfined gas mixing in anaerobic digestion." In: *Computers & Fluids* 180 (2019), pp. 11–21. DOI: 10.1016/j.compfluid.2018.12.008.

[170] Bretl, C., Trunk, R., Nirschl, H., Thäter, G., Dorn, M., and Krause, M. J. "Preliminary Study of Particle Settling Behaviour by Shape Parameters via Lattice Boltzmann Simulations." In: *High Performance Computing in Science and Engineering '20*. Cham, Switzerland, 2021, pp. 245–259. DOI: 10.1007/978-3-030-80602-6_16.

[171] Marquardt, J. E., Eysel, B., Sadric, M., Rauh, C., and Krause, M. J. *Potential for damage to fruits during transport through cross-section constrictions*. 2024. DOI: 10.48550/arXiv.2408.09167. URL: https://arxiv.org/abs/2408.09167.

[172] Schäfer, M., Turek, S., Durst, F., Krause, E., and Rannacher, R. "Benchmark Computations of Laminar Flow Around a Cylinder." In: *Notes on Numerical Fluid Mechanics (NNFM)*. Braunschweig/Wiesbaden, Germany, 1996, pp. 547–566. DOI: 10.1007/978-3-322-89849-4_39.

[173] Latt, J., Chopard, B., Malaspinas, O., Deville, M., and Michler, A. "Straight velocity boundaries in the lattice Boltzmann method." In: *Phys. Rev. E* 77.5 (2008). Publisher: American Physical Society, p. 056703. DOI: 10.1103/PhysRevE.77.056703.

[174] Ten Cate, A., Nieuwstad, C. H., Derksen, J. J., and Van den Akker, H. E. A. "Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity." In: *Physics of Fluids* 14.11 (2002), pp. 4012–4025. DOI: 10.1063/1.1512918.

[175] Hayakawa, Y. and Oguchi, T. "Evaluation of gravel sphericity and roundness based on surface-area measurement with a laser scanner." In: *Computers &; Geosciences* 31.6 (2005), pp. 735–741. DOI: 10.1016/j.cageo.2005.01.004.

[176] Hofmann, H. J. "Grain-shaped indices and isometric graphs." In: *Journal of Sedimentary Research* 64.4a (1994), pp. 916–920. DOI: 10.1306/D4267F0A-2B26-11D7-8648000102C1865D.

[177] Pearson, K. "VII. Mathematical contributions to the theory of evolution.—III. Regression, heredity, and panmixia." In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 187 (1896), pp. 253–318. DOI: 10.1098/rsta.1896.0007.

[178] Konstandopoulos, A. G., Skaperdas, E., Warren, J., and Allansson, R. "Optimized Filter Design and Selection Criteria for Continuously Regenerating Diesel Particulate Traps." In: *International Congress & Exposition*. 1999. DOI: 10.4271/1999-01-0468.

[179] Dittler, A. "Ash Transport in Diesel Particle Filters." In: *SAE 2012 International Powertrains, Fuels & Lubricants Meeting*. 2012. DOI: 10.4271/2012-01-1732.

[180] Kimura, K., Lynskey, M., Corrigan, E. R., Hickman, D. L., Wang, J., Fang, H. L., and Chatterjee, S. "Real World Study of Diesel Particulate Filter Ash Accumulation in Heavy-Duty Diesel Trucks." In: *Powertrain & Fluid Systems Conference and Exhibition*. 2006. DOI: 10.4271/2006-01-3257.

[181] Herschel, W. H. and Bulkley, R. "Konsistenzmessungen von Gummi-Benzollösungen." In: *Kolloid-Zeitschrift* 39.4 (1926), pp. 291–300. DOI: 10.1007/BF01432034.

[182] Bouzidi, M., Firdaouss, M., and Lallemand, P. "Momentum transfer of a Boltzmann-lattice fluid with boundaries." In: *Physics of Fluids* 13.11 (2001), pp. 3452–3459. DOI: 10.1063/1.1399290.

[183] Metzner, A. B. and Reed, J. C. "Flow of non-newtonian fluids—correlation of the laminar, transition, and turbulent-flow regions." In: *AIChE J.* 1.4 (1955), pp. 434–440. DOI: 10.1002/aic.690010409.

[184] Faust, E., Schlüter, A., Müller, H., Steinmetz, F., and Müller, R. "Dirichlet and Neumann boundary conditions in a lattice Boltzmann method for elastodynamics." In: *Computational Mechanics* (2023), pp. 1–23. DOI: 10.1007/s00466-023-02369-w.

[185] Maquart, T., Noël, R., Courbebaisse, G., and Navarro, L. "Toward a Lattice Boltzmann Method for Solids—Application to Static Equilibrium of Isotropic Materials." In: *Applied Sciences* 12.9 (2022). DOI: 10.3390/app12094627.

[186] Gaedtke, M., Wachter, S., Rädle, M., Nirschl, H., and Krause, M. J. "Application of a lattice Boltzmann method combined with a Smagorinsky turbulence model to spatially resolved heat flux inside a refrigerated vehicle." In: *Computers & Mathematics with Applications* 76.10 (2018), pp. 2315–2329. DOI: https://doi.org/10.1016/j.camwa.2018.08.018.

[187] Simonis, S., Oberle, D., Gaedtke, M., Jenny, P., and Krause, M. J. "Temporal large eddy simulation with lattice Boltzmann methods." In: *Journal of Computational Physics* 454 (2022), p. 110991. DOI: https://doi.org/10.1016/j.jcp.2022.110991.

Appendix

A List of Publications

Peer-Reviewed Publications

- **Marquardt, J. E.**, Arlt, C.-R., Trunk, R., Franzreb, M., and Krause, M. J. “Numerical and experimental examination of the retention of magnetic nanoparticles in magnetic chromatography.” In: *Computers & Mathematics with Applications* 89 (2021), pp. 34–43. DOI: 10.1016/j.camwa.2021.02.010.
- **Marquardt, J. E.**, Römer, U. J., Nirschl, H., and Krause, M. J. “A discrete contact model for complex arbitrary-shaped convex geometries.” In: *Particuology* 80 (2023), pp. 180–191. DOI: 10.1016/j.partic.2022.12.005.
- **Marquardt, J. E.**, Hafen, N., and Krause, M. J. “A novel particle decomposition scheme to improve parallel performance of fully resolved particulate flow simulations.” In: *Journal of Computational Science* 78 (2024), p. 102263. DOI: 10.1016/j.jocs.2024.102263.
- **Marquardt, J. E.**, Hafen, N., and Krause, M. J. “A novel model for direct numerical simulation of suspension dynamics with arbitrarily shaped convex particles.” In: *Computer Physics Communications* 304 (2024), p. 109321. DOI: 10.1016/j.cpc.2024.109321.
- **Marquardt, J. E.** and Krause, M. J. “A Review of the Homogenized Lattice Boltzmann Method for Particulate Flow Simulations: From Fundamentals to Applications.” In: *Powders* 3.4 (2024), pp. 500–530. DOI: 10.3390/powders3040027.

- Trunk, R., **Marquardt, J.**, Thäter, G., Nirschl, H., and Krause, M. J. "Towards the simulation of arbitrarily shaped 3D particles using a homogenised lattice Boltzmann method." In: *Computers & Fluids* 172 (2018), pp. 621–631. DOI: 10.1016/j.compfluid.2018.02.027.
- Krause, M. J., Kummerländer, A., Avis, S. J., Kusumaatmaja, H., Dapelo, D., Klemens, F., Gaedtke, M., Hafen, N., Mink, A., Trunk, R., **Marquardt, J. E.**, Maier, M.-L., Haussmann, M., and Simonis, S. "OpenLB—Open source lattice Boltzmann code." In: *Computers & Mathematics with Applications* 81 (2021), pp. 258–288. DOI: 10.1016/j.camwa.2020.04.033.
- Jeßberger, J., **Marquardt, J. E.**, Heim, L., Mangold, J., Bukreev, F., and Krause, M. J. "Optimization of a Micromixer with Automatic Differentiation." In: *Fluids* 7.5 (2022). DOI: 10.3390/fluids7050144.
- Hafen, N., **Marquardt, J. E.**, Dittler, A., and Krause, M. J. "Simulation of Particulate Matter Structure Detachment from Surfaces of Wall-Flow Filters for Elevated Velocities Applying Lattice Boltzmann Methods." In: *Fluids* 8.3 (2023), p. 99. DOI: 10.3390/fluids8030099.
- Hafen, N., **Marquardt, J. E.**, Dittler, A., and Krause, M. J. "Simulation of Dynamic Rearrangement Events in Wall-Flow Filters Applying Lattice Boltzmann Methods." In: *Fluids* 8.7 (2023), p. 213. DOI: 10.3390/fluids8070213.

Other Publications

- Kummerländer, A., Avis, S., Kusumaatmaja, H., Bukreev, F., Crocoll, M., Dapelo, D., Großmann, S., Hafen, N., Ito, S., Jeßberger, J., Kummer, E., **Marquardt, J. E.**, Mödl, J., Pertzel, T., Prinz, F., Raichle, F., Sadric, M., Schecher, M., Teutscher, D., Simonis, S., and Krause, M. J. *OpenLB User Guide: Associated with Release 1.6 of the Code*. 2023. DOI: 10.48550/arXiv.2307.11752. URL: https://doi.org/10.48550/arXiv.2307.11752.
- **Marquardt, J. E.**, Eysel, B., Sadric, M., Rauh, C., and Krause, M. J. *Potential for damage to fruits during transport through cross-section constrictions*. 2024. DOI: 10.48550/arXiv.2408.09167. URL: https://arxiv.org/abs/2408.09167.

Conference Talks

- **Marquardt, J. E.**, Trunk, R., Nirschl, H., and Krause, M. J. "Numerische Studie des Segré-Silberberg Effekt mit der Homogenised Lattice Boltzmann

Methode." In: *Jahrestreffen der ProcessNet-Fachgruppen Computational Fluid (CFD) Dynamics und Mehrphasenströmungen (MPH)*. Online, 2021.

- **Marquardt, J. E.**, Hafen, N., Römer, U. J., Nirschl, H., and Krause, M. J. "Modeling and simulation of particulate fluid flows with complex shapes and four-way coupling using the homogenized lattice Boltzmann method." In: *PARTEC*. Nuremberg, Germany, 2023.

Conference Posters

- **Marquardt, J. E.**, Hafen, N., Trunk, R., Nirschl, H., and Krause, M. J. "SPP 2045: Modelling and Simulation of the Shape-Dependent Settling Behavior of Particles." In: *PARTEC*. Nuremberg, Germany, 2023.
- **Marquardt, J. E.** and Krause, M. J. "Direct numerical simulation of suspension dynamics with arbitrarily shaped convex particles." In: *Jahrestreffen Dechema-Fachgruppen Mehrphasenströmung (MPH), Computational Fluid Dynamics (CFD) und Aerosoltechnik (AT)*. Bremen, Germany, 2024.

Software Releases

- Krause, M., Avis, S., Kusumaatmaja, H., Dapelo, D., Gaedtke, M., Hafen, N., Haußmann, M., Jeppener-Haltenhoff, J., Kronberg, L., Kummerländer, A., **Marquardt, J. E.**, Pertzel, T., Simonis, S., Trunk, R., Wu, M., and Zarth, A. *OpenLB Release 1.4: Open Source Lattice Boltzmann Code*. Version 1.4. 2020. DOI: 10.5281/zenodo.4279263.
- Kummerländer, A., Avis, S., Kusumaatmaja, H., Bukreev, F., Dapelo, D., Großmann, S., Hafen, N., Holeksa, C., Husfeldt, A., Jeßberger, J., Kronberg, L., **Marquardt, J. E.**, Mödl, J., Nguyen, J., Pertzel, T., Simonis, S., Springmann, L., Suntoyo, N., Teutscher, D., Zhong, M., and Krause, M. J. *OpenLB Release 1.5: Open Source Lattice Boltzmann Code*. Version 1.5. 2022. DOI: 10.5281/zenodo.6469606.
- Kummerländer, A., Avis, S., Kusumaatmaja, H., Bukreev, F., Crocoll, M., Dapelo, D., Hafen, N., Ito, S., Jeßberger, J., **Marquardt, J. E.**, Mödl, J., Pertzel, T., Prinz, F., Raichle, F., Schecher, M., Simonis, S., Teutscher, D., and Krause, M. J. *OpenLB Release 1.6: Open Source Lattice Boltzmann Code*. Version 1.6. 2023. DOI: 10.5281/zenodo.7773497.

- Kummerländer, A., Bingert, T., Bukreev, F., Czelusniak, L. E., Dapelo, D., Hafen, N., Heinzelmann, M., Ito, S., Jeßbberger, J., Kusumaatmaja, H., **Marquardt**, **J. E.**, Rennick, M., Pertzel, T., Prinz, F., Sadric, M., Schecher, M., Simonis, S., Sitter, P., Teutscher, D., Zhong, M., and Krause, M. J. *OpenLB Release 1.7: Open Source Lattice Boltzmann Code*. Version 1.7.0. 2024. DOI: 10.5281/zenodo.10684609.

B List of Application Cases

The following lists all presented simulation cases along with the related commit hashes. They can be found on the main branch of OpenLBs repository hosted at gitlab.com/openlb/olb.

1. Two-way coupled hindered settling of spherical particles

   Description: see Chapter 2
   Commit hash:
   `87e497ce6003e962155d1defd88936cbd24001ae`
   Case folder: `apps/jan/collisions/hinderedSettling`
   Date: 16/05/2023
   Released state: Version 1.7

2. Dry contact — Calculation of contact forces

   Description: see Chapter 3
   Commit hash:
   `372a30f52b8f9fc770bcccea81fca5f6bc1200c8`
   Case folder:
   `apps/jan/collisions/forceCalculationTests`
   Date: 22/11/2022
   Released state: Version 1.5

3. Dry contact — Cylinder-wall impact

   Description: see Chapter 3
   Commit hash:
   `372a30f52b8f9fc770bcccea81fca5f6bc1200c8`
   Case folder: `apps/jan/collisions/cylWallImpactTest`
   Date: 22/11/2022
   Released state: Version 1.5

4. Particle rebound in a viscous fluid

   Description: see Chapter 3
   Commit hash:
   `372a30f52b8f9fc770bcccea81fca5f6bc1200c8`
   Case folder: `apps/jan/collisions/bouncingSphere`
   Date: 22/11/2022
   Released state: Version 1.5

5. Four-way coupled hindered settling of spherical and cubic particles

   Description: see Chapter 5
   Commit hash:
   `4a81a62eaba9ccf53ed947ad192558dbb1f09335`
   Case folder: `apps/jan/collisions/hinderedSettling`
   Date: 05/10/2023
   Released state: Version 1.7

6. Fruit damage potential studies — Pure fluid flow simulation

   Description: see Chapter 6
   Commit hash:
   `849fd1549bf88d9792bf8d13f424ae439ecb2e33`
   Case folder: `apps/jan/nonNewtonian/hbPipe3d`
   Date: 29/01/2024
   Released state: Version 1.7

7. Fruit damage potential studies — Particulate flow simulation

   Description: see Chapter 6
   Commit hash:
   `849fd1549bf88d9792bf8d13f424ae439ecb2e33`
   Case folder: `apps/jan/nonNewtonian/hbPipe3dParticle`
   Date: 29/01/2024
   Released state: Version 1.7

8. Fruit damage potential studies — Data analysis

   Description: see Chapter 6
   Commit hash:
   `dfbc161064663159614c3a31c7a1cab6896cac76`
   Case folder: `apps/jan/nonNewtonian/hbPipe3dParticle`
   Date: 25/04/2024
   Released state: Version 1.7

C Acronyms

BFS breadth-first search

BGK Bhatnagar-Gross-Krook

CAD computer-aided design

CFD computational fluid dynamics

CI continuous integration

CSE common subexpression elimination

CT computed tomography

DEM discrete element method

DNS direct numerical simulation

EDM exact difference method

EOC experimental order of convergence

GPU graphics processing unit

HLBM homogenized lattice Boltzmann method

IBM immersed boundary method

LBE lattice Boltzmann equation

LBM lattice Boltzmann method

LES large eddy simulation

MEA momentum exchange algorithm

MLUPs million lattice site updates per second

PSM partially saturated method

PU processing unit

SDF signed distance function

SGS subgrid scale

STL stereolithography

VANSE volume-averaged Navier–Stokes equations