A novel particle decomposition scheme to improve parallel performance of fully resolved particulate flow simulations

Jan E. Marquardt [a,b,*], Nicolas Hafen [a,b], Mathias J. Krause [a,b,c]

[a] *Lattice Boltzmann Research Group, Karlsruhe Institute of Technology, Straße am Forum 8, Karlsruhe, 76131, Germany*
[b] *Institute for Mechanical Process Engineering and Mechanics, Karlsruhe Institute of Technology, Straße am Forum 8, Karlsruhe, 76131, Germany*
[c] *Institute for Applied and Numerical Mathematics, Karlsruhe Institute of Technology, Englerstraße 2, Karlsruhe, 76131, Germany*

ARTICLE INFO

ABSTRACT

This study addresses the challenge of simulating realistic particle systems by proposing a novel particle decomposition scheme that improves the parallel performance of surface resolved particle simulations. Realistic particle systems often involve large numbers of particles and complex particle shapes. The resulting need to account for shape factors requires the inclusion of even more particles to obtain statistically meaningful results. However, the computational cost increases with the number of particles, making efficient parallelization crucial. Therefore, the proposed scheme aims to improve the scalability by optimizing the communication and data management between processors. Through hindered settling experiments, the applicability and performance of the novel particle decomposition scheme are thoroughly investigated using the homogenized lattice Boltzmann method. The results show that the proposed method significantly improves the performance, especially in scenarios with a large number of particles, by reducing communication constraints and improving scalability. This research contributes to the advancement of computational methods for efficient study of complex particle systems and provides valuable insights for future developments in this field.

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 [1–5]. These early studies provided valuable insights into the collective behavior of particles and laid the foundation for further investigations of dilute [6] and concentrated suspensions at moderate Reynolds numbers [7,8]. Subsequently, the research expanded to include the formation and dynamics of particle clusters during hindered settling [9,10]. 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 [11], fibers [12], and irregular sand grains [13]. 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

---

* Corresponding author at: Institute for Mechanical Process Engineering and Mechanics, Karlsruhe Institute of Technology, Straße am Forum 8, Karlsruhe, 76131, Germany.
E-mail address: jan.marquardt@kit.edu (J.E. Marquardt).

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 [14] and can represent arbitrary geometries using the glued-sphere technique [15] or similarly by combining other convex shapes [16]. 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 [17,18] or model only the drag coefficient without back-coupling from the particles to the fluid [19]. However, the back-coupling is of enormous importance, especially at high particle volume fractions [20].

The immersed boundary method (IBM) is another well-known option to consider non-spherical particles, since it resolves surfaces as Lagrange points [21]. 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 [22,23], which has also been addressed by other LBM-based methods [24]. However, the frequent interpolations between particle and fluid points are computationally expensive.

The partially saturated method (PSM), originally introduced by Noble and Torczynski [25] 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 [26]. Subsequently, various new derivatives have been introduced [27,28], including the homogenized lattice Boltzmann method (HLBM) [29–31]. 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 [32]. In addition, HLBM has been applied to the simulation of cubic disk-shaped particles in wall flow filters [33–36] and another PSM derivative to the filtration of irregular airborne particles [37]. The potential applications are further extended by a compatible contact model that is suitable for arbitrary convex shapes [38]. 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 introduces the models used to consider the fluid and the particles, while Section 3 discusses the numerical methods used to solve the model system, followed by the proposed domain decomposition scheme in Section 4 and its application to hindered settling in Section 5. Finally, Section 6 provides a summary and conclusion.

2. Modeling

2.1. Fluid

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

$$\frac{\partial \boldsymbol{u}_f}{\partial t} + \left(\boldsymbol{u}_f \cdot \nabla\right) \boldsymbol{u}_f - \nu \Delta \boldsymbol{u}_f + \frac{1}{\rho_f} \nabla p = \boldsymbol{F}_f,$$
$$\nabla \cdot \boldsymbol{u}_f = 0, \tag{1}$$

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

2.2. Particle

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

$$m_p \frac{\partial \boldsymbol{u}_p}{\partial t} = \boldsymbol{F}_p \tag{2}$$

and rotation by

$$\boldsymbol{I}_p \frac{\partial \boldsymbol{\omega}_p}{\partial t} + \boldsymbol{\omega}_p \times \left(\boldsymbol{I}_p \cdot \boldsymbol{\omega}_p\right) = \boldsymbol{T}_p. \tag{3}$$

Here, m_p, $\boldsymbol{I}_p$, $\boldsymbol{u}_p$, $\boldsymbol{\omega}_p$ are the mass, moment of inertia, velocity, and angular velocity of the particle. $\boldsymbol{F}_p$ and $\boldsymbol{T}_p$ are the sum of the acting forces and torques affecting the particle motion, which may include the hydrodynamic forces and torques mentioned in Section 3.2 or contact treatment results [38,39]. Above, the subscript p indicates that the quantities refer to the particle's center of mass.

3. Numerical methods

3.1. Lattice Boltzmann method

In this work, the Navier–Stokes equations for incompressible flows are solved using the lattice Boltzmann method (LBM) [26,40,41]. Note that all values in Section 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 [26,40]. For the studies in this paper, we choose the D3Q19, which discretizes the three-dimensional space and contains 19 discrete velocities

$$\boldsymbol{c}_i = \begin{cases} (0,0,0), & \text{if } i = 0 \\ (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1), & \text{if } i = 1, \dots, 6 \\ (\pm 1, \pm 1, 0), (\pm 1, 0, \pm 1), (0, \pm 1, \pm 1), & \text{if } i = 7, \dots, 18 \end{cases}. \tag{4}$$

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

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

$$f_i^*(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) + \Omega_i(\boldsymbol{x}, t) + S_i(\boldsymbol{x}, t). \tag{5}$$

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

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i \Delta t, t + \Delta t) = f_i^*(\boldsymbol{x}, t), \tag{6}$$

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 [42]

$$\Omega_i(\boldsymbol{x}, t) = -\frac{1}{\tau}(f_i(\boldsymbol{x}, t) - f_i^{\text{eq}}(\rho_{\text{f}}, \boldsymbol{u}_{\text{f}})), \tag{7}$$

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

$$\tau = \left(3\nu + \frac{1}{2}\right). \tag{8}$$

The Maxwell–Boltzmann distribution, quantifying the equilibrium state is given by [43]

$$f_i^{\text{eq}}(\rho_{\text{f}}, \boldsymbol{u}_{\text{f}}) = w_i \rho_{\text{f}} \left(1 + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}_{\text{f}}}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}_{\text{f}})^2}{2c_s^4} - \frac{\boldsymbol{u}_{\text{f}} \cdot \boldsymbol{u}_{\text{f}}}{2c_s^2}\right). \tag{9}$$

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

$$w_i = \begin{cases} 1/3, & \text{if } i = 0 \\ 1/18, & \text{if } i = 1, \dots, 6 \\ 1/36, & \text{if } i = 7, \dots, 18 \end{cases}, \tag{10}$$

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 [44,45] is used exclusively in this work.

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 [29–31]. Using the trigonometric level set function [38]

$$B(\boldsymbol{x}, t) = \begin{cases} 1, & \text{if } d_s \leq -\varepsilon/2 \\ \cos^2\left(\frac{\pi}{2}\left(\frac{d_s}{\varepsilon} + \frac{1}{2}\right)\right), & \text{if } \varepsilon/2 > d_s > -\varepsilon/2 \\ 0, & \text{if } d_s \geq \varepsilon/2 \end{cases}, \tag{11}$$

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. [29]. 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

$$\Delta\boldsymbol{u}_{\text{f}}(\boldsymbol{x}, t) = B(\boldsymbol{x}, t)\left(\boldsymbol{u}_{\text{p}}(\boldsymbol{x}, t) - \boldsymbol{u}_{\text{f}}(\boldsymbol{x}, t)\right), \tag{12}$$

where $\boldsymbol{u}_{\text{p}}(\boldsymbol{x}) = \boldsymbol{u}_{\text{p}}(\boldsymbol{X}_{\text{p}}) + \boldsymbol{\omega}_{\text{p}} \times (\boldsymbol{x} - \boldsymbol{X}_{\text{p}})$ is the particle's velocity at position $\boldsymbol{x}$, and $\boldsymbol{X}_{\text{p}}$ is the particle's center of mass. According to previous studies [31], the best results are obtained by using an adapted exact difference method (EDM) [46]. This method introduces the following source term in Eq. (5)

$$S_i(\boldsymbol{x}, t) = f_i^{\text{eq}}(\rho_{\text{f}}, \boldsymbol{u}_{\text{f}} + \Delta\boldsymbol{u}_{\text{f}}) - f_i^{\text{eq}}(\rho_{\text{f}}, \boldsymbol{u}_{\text{f}}). \tag{13}$$

To couple from the fluid to the particle, we use the momentum exchange algorithm (MEA) by Wen et al. [47] to calculate the hydrodynamic force

$$\boldsymbol{F}_{\text{h}}(\boldsymbol{x}, t) = \sum_i (\boldsymbol{c}_i - \boldsymbol{u}_{\text{p}}(\boldsymbol{x}, t)) f_i(\boldsymbol{x} + \boldsymbol{c}_i \Delta t, t) + (\boldsymbol{c}_i + \boldsymbol{u}_{\text{p}}(\boldsymbol{x}, t)) f_{\bar{i}}(\boldsymbol{x}, t). \tag{14}$$

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 $\boldsymbol{x}_{\text{b}}$. Thus the total force is [31]

$$\boldsymbol{F}_{\text{p}}(t) = \sum_{\boldsymbol{x}_{\text{b}}} \boldsymbol{F}_{\text{h}}(\boldsymbol{x}_{\text{b}}, t), \tag{15}$$

and the torque is given by

$$\boldsymbol{T}_{\text{p}}(t) = \sum_{\boldsymbol{x}_{\text{b}}} (\boldsymbol{x}_{\text{b}} - \boldsymbol{X}_{\text{p}}) \times \boldsymbol{F}_{\text{h}}(\boldsymbol{x}_{\text{b}}, t). \tag{16}$$

Note that while an applicable contact model exists [38], 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.

4. Particle decomposition scheme

4.1. Background

In the context of the LBM, parallelization is typically implemented using a block-based approach [40]. 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.

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.

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.
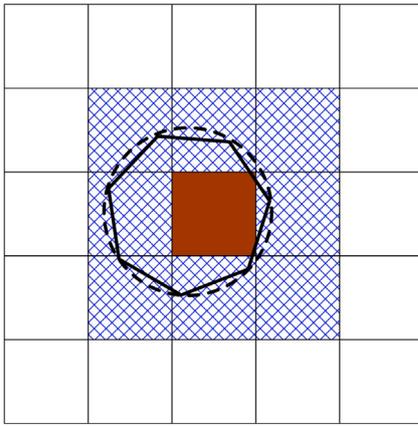
Fig. 1. Representation of a heptagonal particle (solid black) and its circumferential radius (dashed black) in a multi-block domain consisting of the responsible center block (brown) and neighboring blocks (blue cross-hatching).

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. [48]. 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 Fig. 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 Fig. 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.

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 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 Fig. 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.
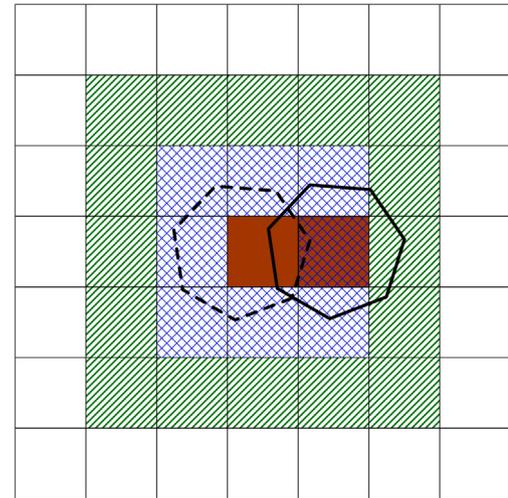
Fig. 2. Illustration of the position of a heptagonal structure at the current (dashed black) and next (solid black) time steps within a multi-block domain. The brown middle block represents the currently responsible block, while the adjacent blocks with blue cross-hatching indicate the first-level neighborhood. 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.

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.

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 Fig. 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 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 Fig. 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.

4.2.5. Time step algorithm

To provide a clear and organized presentation, Algorithm 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 [45]. The algorithm follows a specific sequence that is consistent with the previously discussed methods and ensures the necessary order of execution.
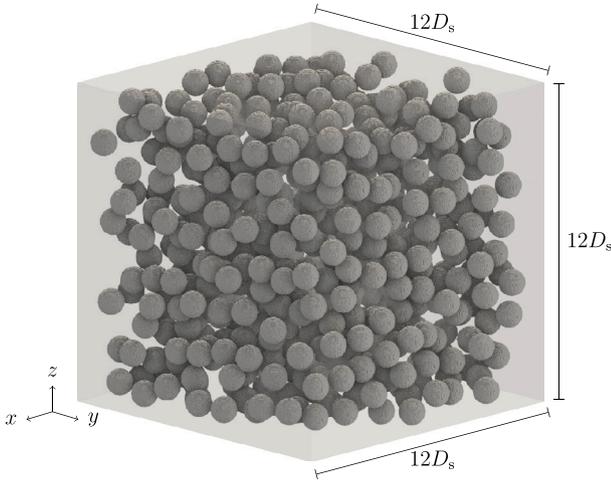
Fig. 3. Simulation domain with the spherical particles on their initial position using a resolution of $N = 27$ cells per sphere diameter and particle volume fraction of about 0.2.
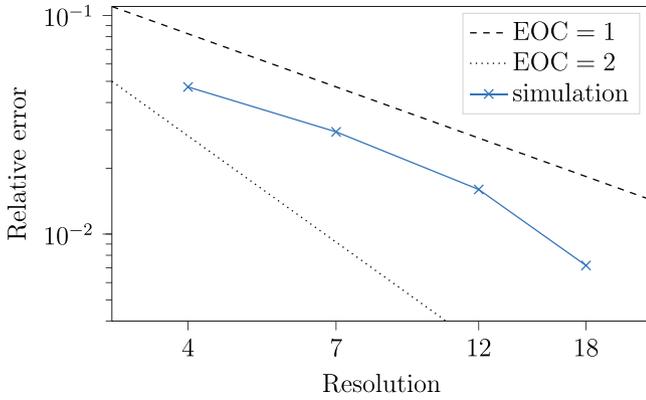
Fig. 4. Relative error in L^2 norm versus the resolution of the sphere's diameter N. The reference uses the resolution $N = 27$.

Algorithm 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 4.2.2

  Apply external forces;   ▷ Such as gravity

  Solve equations of motion;

  Evaluate particle assignment;   ▷ See Section 4.2.3

  Communicate data and assignment;   ▷ See Section 4.2.4

  Couple particles to fluid;   ▷ Using the EDM

  Perform collision and streaming;

  Increase time step;

end

---

5. Application to hindered settling

In the following, we evaluate the application of the proposed scheme to hindered settling [1]. 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 [2], use a power-law model such as

$$\bar{u}_p = u^*(1 - \phi_p)^n, \tag{17}$$

with the predicted average settling velocity $\bar{u}_p$, a reference terminal velocity of a single particle in the considered domain u^*, the particle volume fraction ϕ_p and an expansion index

$$n = \begin{cases} 4.65, & \text{if } Re < 0.2 \\ 4.35\,Re^{-0.03}, & \text{if } 0.2 \leq Re < 1 \\ 4.45\,Re^{-0.1}, & \text{if } 1 \leq Re < 500 \\ 2.39, & \text{if } 500 \leq Re \end{cases}. \tag{18}$$

Here, n depends on the Reynolds number $Re = u^* D_s/\nu$. Garside and Al-Dibouni [4] suggest

$$n = \frac{5.1 + 0.27\,Re^{0.9}}{1 + 0.1\,Re^{0.9}}, \tag{19}$$

which has shown a superior accuracy [49]. Barnea and Mizrahi [3] propose a different model, in which the average settling velocity is independent of Re and is given by

$$\bar{u}_p = u^* \frac{1 - \phi_p}{(1 + \phi_p^{1/3}) \exp \frac{5\phi_p}{3(1-\phi_p)}}. \tag{20}$$

For validation purposes, all three correlations are used below.

In the creeping flow regime, the reference velocity reads

$$u_{St}^* = \frac{g D_s^2}{18\nu} \left(\frac{\rho_p - \rho_f}{\rho_f} \right), \tag{21}$$

where $g = 9.80665 \text{ m s}^{-2}$ 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

$$u^* = \sqrt{\frac{4\,g D_s}{3C_d} \left(\frac{\rho_p - \rho_f}{\rho_f} \right)}, \tag{22}$$

with the drag coefficient C_d that we compute using the approximation by Schiller and Neumann [50]

$$C_d = \frac{24}{Re} \left(1 + 0.15\,Re^{0.687} \right), \tag{23}$$

which is valid for $Re < 800$.

As is common in studies of hindered settling [10,49], we use two dimensionless parameters to describe the setup below. The first one is the particle to fluid density ratio defined as ρ_p/ρ_f and the second is the Archimedes number

$$Ar = \frac{g D_s^3 \frac{\rho_p - \rho_f}{\rho_f}}{\nu^2}. \tag{24}$$

5.1. Simulation setup

To numerically study the above, we employ spherical particles with a diameter $D_s = 2$ mm at random positions in a cubic domain with an edge length of $12 D_s$ and periodic boundaries on each side, which is filled with a fluid with a density $\rho_f = 1000 \text{ kg m}^{-3}$. The setup is exemplified in Fig. 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 $10 D_s$ [10,49]. 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 $F_g = (\rho_f - \rho_p)V_s g$. Here, $V_s = \pi D_s^3/6$ is the sphere's volume. As in previous numerical investigations of hindered settling [9,10,49], 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 [51] with a superficial velocity equal to the current average settling velocity $\bar{u}_p$.
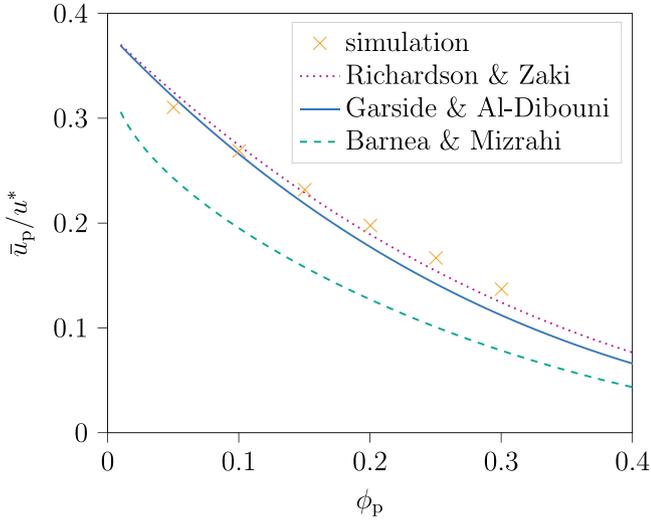
Fig. 5. Ratio of average settling velocity to single particle settling velocity $\bar{u}_p/u^*$ versus the particle volume fraction ϕ_p, showing current simulation results and the correlations by Richardson and Zaki [2], Garside and Al-Dibouni [4], as well as Barnea and Mizrahi [3].

In our studies, the Archimedes number is fixed at $Ar = 1500$ and the density ratio at $\rho_p/\rho_f = 1.3$. 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 $t^* = D_s/u^*_{St}$. Due to the initial random packing, the startup time is significantly reduced, nonetheless, we start averaging the velocities to obtain $\bar{u}_p$ after a time of $30t^*$, leaving a period of $270t^*$ that ensures statistically averaged results.

5.2. Grid independence study

In this section, we delve into a grid independence study using a particle volume fraction of $\phi_p = 0.1$ by systematically exploring various grid resolutions $N \in \{5, 7, 12, 18\}$. 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 [31]. 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 Fig. 4, which illustrates the relative error using the L^2 norm [40] 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 [31] and suggests that increasing the grid resolution leads to a proportional decrease in error. It is likely due to the staircase approximation of the curved boundary [40,52]. 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 [31]. This adjustment is due to the presence of multiple particles and their interactions through the fluid.

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 Fig. 5. For comparison, we additionally add the correlation by Richardson and Zaki [2] as a dotted purple line, Garside and Al-Dibouni [4] as a blue line, and the correlation by Barnea and Mizrahi [3] 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 explicit 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 [31]. 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 [38,39] 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.

Fig. 6 visualizes the fluid velocity and the particles for $\phi_p \approx 0.3$ 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 [10], 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.

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.

Fig. 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 Figs. 7(a) and 7(b), a resolution of 18 in Figs. 7(c) and 7(d), and a resolution of 27 in Figs. 7(e) and 7(f). In each scenario, the number of particles ranges from 166 to 991.

The performance plots in Fig. 7 reveal several important observations: Firstly, the novel method exhibits in general higher MLUPs compared to the 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

(a) $t^* \approx 49.97$ (b) $t^* \approx 299.82$
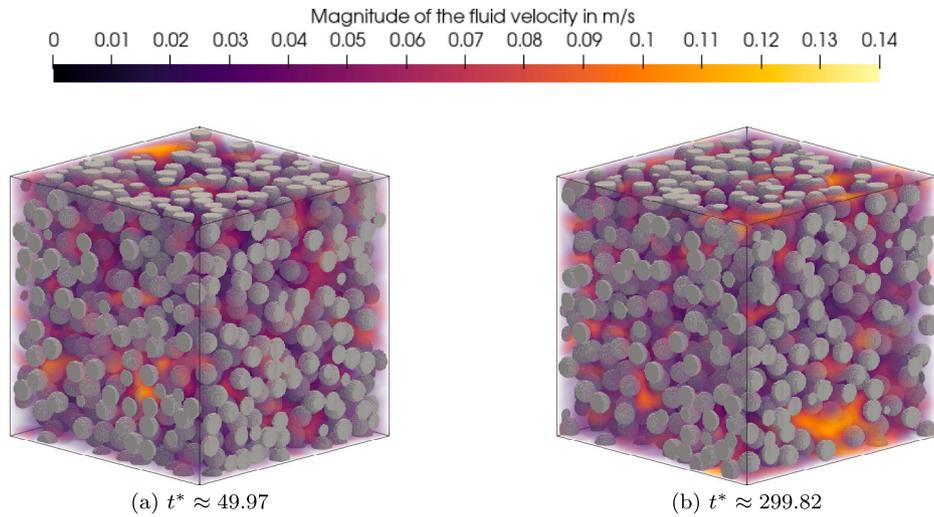
Fig. 6. Particles and fluid velocity at different normalized times t^* for $Ar = 1500$ and $\phi_p \approx 0.3$.

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 Fig. 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 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.

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.

Nomenclature

Acronyms

BGK	Bhatnagar–Gross–Krook
DEM	discrete element method
EDM	exact difference method
EOC	experimental order of convergence
HLBM	homogenized lattice Boltzmann method
IBM	immersed boundary method
LBM	lattice Boltzmann method
MEA	momentum exchange algorithm
MLUPs	million lattice site updates per second
PSM	partially saturated method

Roman Symbols

B	weighting factor
c	discrete velocity
c_s	lattice speed of sound
D	diameter
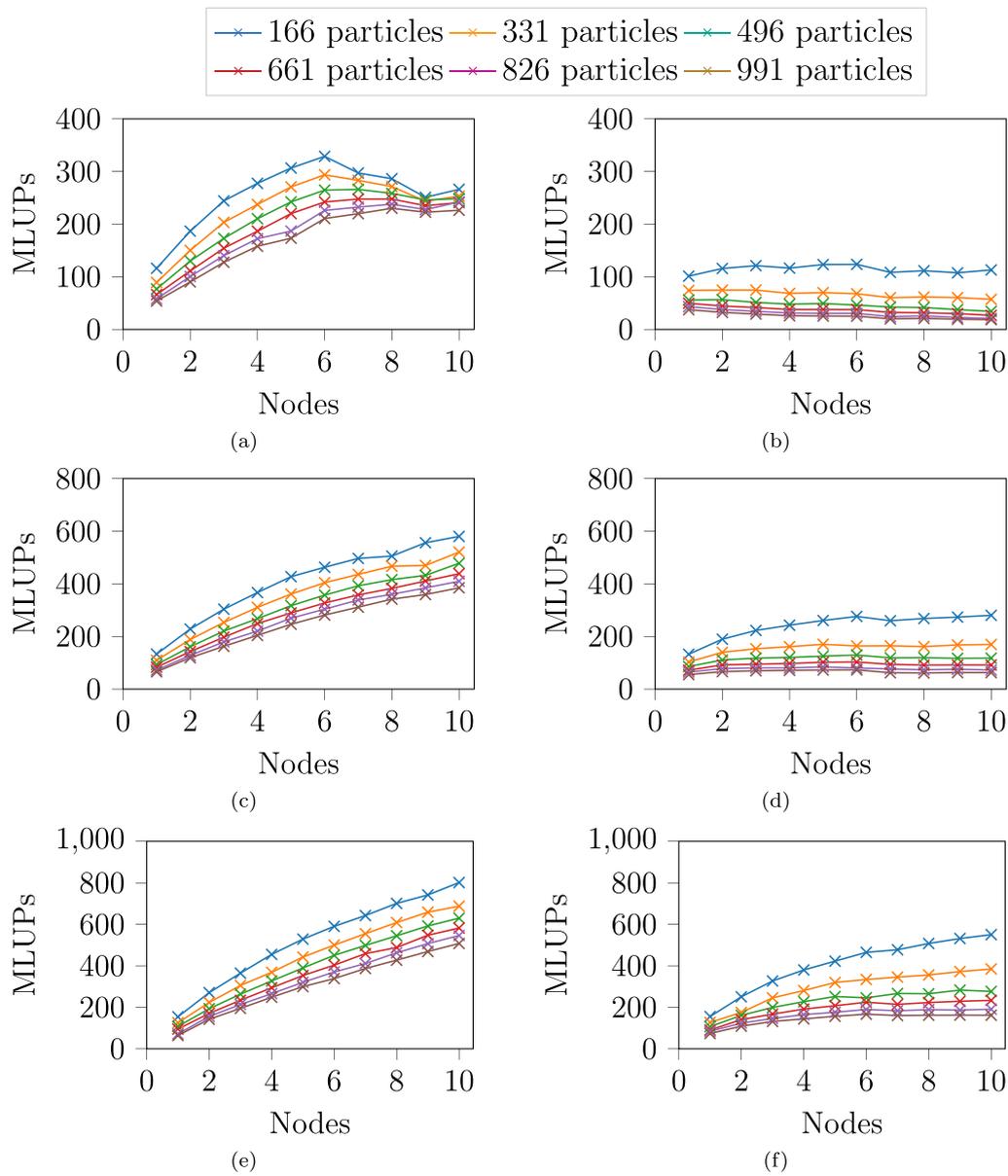
Fig. 7. Comparison of the resulting MLUPs versus the number of employed nodes using the new particle decomposition scheme (left) and the conventional decomposition (right) for the resolutions $N = 12$ (**a** and **b**), $N = 18$ (**c** and **d**) and $N = 27$ (**e** and **f**) and different particle numbers.

d_s	signed distance		X	center of mass
F	total force		x	position
F_g	combination of weight and buoyancy			
f	particle population		**Greek Symbols**	
f^*	post-collision particle population			
g	standard gravity		Δt	time step size
I	moment of inertia		Δx	grid spacing
m	mass		ε	size of the smooth boundary
N	resolution		ν	kinematic viscosity
n	expansion index		ρ	density
Re	Reynolds number		τ	relaxation time
S	source term		ϕ	volume fraction
T	total torque		Ω	collision operator
t	time		ω	angular velocity
u	velocity			
u^*	reference settling velocity of a single sphere		**Subscripts**	
V	volume			
w	weight for the equilibrium distribution calculation		b	refers to positions inside a particle's boundary
			f	refers to the fluid

h	refers to the hydrodynamic force
i	refers to the corresponding discrete velocity
p	refers to the particle's center of mass
s	refers to a sphere
St	refers to the Stokes/creeping flow regime

CRediT authorship contribution statement

Jan E. Marquardt: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Nicolas Hafen:** Conceptualization, Methodology, Project administration, Software, Writing – review & editing. **Mathias J. Krause:** Funding acquisition, Project administration, Resources, Software, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

References

[1] H.H. Steinour, Rate of sedimentation: Nonflocculated suspensions of uniform spheres, Ind. Eng. Chem. 36 (7) (1944) 618–624, http://dx.doi.org/10.1021/ie50415a005.

[2] J.F. Richardson, W.N. Zaki, Sedimentation and fluidisation: Part I, Chem. Eng. Res. Des. 75 (1954) S82–S100, http://dx.doi.org/10.1016/S0263-8762(97)80006-8.

[3] E. Barnea, J. Mizrahi, A generalized approach to the fluid dynamics of particulate systems, Chem. Eng. J. 5 (2) (1973) 171–189, http://dx.doi.org/10.1016/0300-9467(73)80008-5.

[4] J. Garside, M.R. Al-Dibouni, Velocity-voidage relationships for fluidization and sedimentation in solid–liquid systems, Ind. Eng. Chem. Process Des. Develop. 16 (2) (1977) 206–214, citation Key: garside1977velocity.

[5] R. Di Felice, Hydrodynamics of liquid fluidisation, Chem. Eng. Sci. 50 (8) (1995) 1213–1245, http://dx.doi.org/10.1016/0009-2509(95)98838-6.

[6] R. Di Felice, The sedimentation velocity of dilute suspensions of nearly monosized spheres, Int. J. Multiph. Flow 25 (4) (1999) 559–574, http://dx.doi.org/10.1016/S0301-9322(98)00084-6.

[7] A. Hamid, J.J. Molina, R. Yamamoto, Direct numerical simulations of sedimenting spherical particles at non-zero reynolds number, RSC Adv. 4 (96) (2014) 53681–53693, http://dx.doi.org/10.1039/C4RA11025K.

[8] A.A. Zaidi, T. Tsuji, T. Tanaka, Hindered settling velocity & structure formation during particle settling by direct numerical simulation, Procedia Eng. 102 (2015) 1656–1666, http://dx.doi.org/10.1016/j.proeng.2015.01.302.

[9] D.P. Willen, A. Prosperetti, Resolved simulations of sedimenting suspensions of spheres, Phys. Rev. Fluids 4 (1) (2019) 014304, http://dx.doi.org/10.1103/PhysRevFluids.4.014304.

[10] Y. Yao, C.S. Criddle, O.B. Fringer, The effects of particle clustering on hindered settling in high-concentration particle suspensions, J. Fluid Mech. 920 (2021) A40, http://dx.doi.org/10.1017/jfm.2021.470.

[11] Y. Chong, D. Ratkowsky, N. Epstein, Effect of particle shape on hindered settling in creeping flow, Powder Technol. 23 (1) (1979) 55–66, http://dx.doi.org/10.1016/0032-5910(79)85025-1.

[12] T. Jirout, D. Jiroutová, Hindered settling of fiber particles in viscous fluids, Processes 10 (99) (2022) 1701, http://dx.doi.org/10.3390/pr10091701.

[13] M.R. Tomkins, T.E. Baldock, P. Nielsen, Hindered settling of sand grains, Sedimentology 52 (6) (2005) 1425–1432, http://dx.doi.org/10.1111/j.1365-3091.2005.00750.x.

[14] H.P. Zhu, Z.Y. Zhou, R.Y. Yang, A.B. Yu, Discrete particle simulation of particulate systems: A review of major applications and findings, Chem. Eng. Sci. 63 (23) (2008) 5728–5770, http://dx.doi.org/10.1016/j.ces.2008.08.006.

[15] G.T. Nolan, P.E. Kavanagh, Random packing of nonspherical particles, Powder Technol. 84 (3) (1995) 199–205, http://dx.doi.org/10.1016/0032-5910(95)98237-S.

[16] A.D. Rakotonirina, J.-Y. Delenne, F. Radjai, A. Wachs, Grains3D, a flexible DEM approach for particles of arbitrary convex shape—Part III: extension to non-convex particles modelled as glued convex particles, Comput. Part. Mech. 6 (1) (2019) 55–84, http://dx.doi.org/10.1007/s40571-018-0198-3.

[17] L.-C. Qiu, C.-Y. Wu, A hybrid DEM/CFD approach for solid–liquid flows, J. Hydrodyn. 26 (2014) 19–25, http://dx.doi.org/10.1016/S1001-6058(14)60003-2.

[18] R. Sun, H. Xiao, Sedifoam: A general-purpose, open-source CFD–DEM solver for particle-laden flow with emphasis on sediment transport, Comput. Geosci. 89 (2016) 207–219, http://dx.doi.org/10.1016/j.cageo.2016.01.011.

[19] M. Weers, L. Hansen, D. Schulz, B. Benker, A. Wollmann, C. Kykal, H. Kruggel-Emden, A.P. Weber, Development of a model for the separation characteristics of a deflector wheel classifier including particle collision and rebound behavior, Minerals 12 (4) (2022) 480, http://dx.doi.org/10.3390/min12040480.

[20] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, B. Van Wachem, Computational Fluid Dynamics for Engineers, Cambridge University Press, 2011, http://dx.doi.org/10.1017/CBO9781139093590.

[21] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005) 448–476, http://dx.doi.org/10.1016/j.jcp.2005.03.017.

[22] A. Amiri Delouei, S. Karimnejad, F. He, Direct numerical simulation of pulsating flow effect on the distribution of non-circular particles with increased levels of complexity: IB-LBM, Comput. Math. Appl. 121 (2022) 115–130, http://dx.doi.org/10.1016/j.camwa.2022.07.005.

[23] S. Karimnejad, A. Amiri Delouei, M. Nazari, A. Shahmardan, A. Mohamad, Sedimentation of elliptical particles using immersed boundary – lattice Boltzmann method: A complementary repulsive force model, J. Mol. Liq. 262 (2018) 180–193, http://dx.doi.org/10.1016/j.molliq.2018.04.075.

[24] Y. Hu, D. Li, S. Shu, X. Niu, Modified momentum exchange method for fluid-particle interactions in the lattice Boltzmann method, Phys. Rev. E 91 (3) (2015) 033301, http://dx.doi.org/10.1103/physreve.91.033301.

[25] D.R. Noble, J.R. Torczynski, A lattice-Boltzmann method for partially saturated computational cells, Internat. J. Modern Phys. C 09 (08) (1998) 1189–1201, http://dx.doi.org/10.1142/S0129183198001084.

[26] S. Succi, The Lattice Boltzmann Equation: for Fluid Dynamics and Beyond, Oxford University Press, 2001.

[27] M. Haussmann, N. Hafen, F. Raichle, R. Trunk, H. Nirschl, M.J. Krause, Galilean invariance study on different lattice Boltzmann fluid–solid interface approaches for vortex-induced vibrations, Comput. Math. Appl. 80 (2020) 671–691, http://dx.doi.org/10.1016/j.camwa.2020.04.022.

[28] C. Rettinger, U. Rüde, A comparative study of fluid-particle coupling methods for fully resolved lattice Boltzmann simulations, Comput. & Fluids 154 (2017) 74–89, http://dx.doi.org/10.1016/j.compfluid.2017.05.033.

[29] M.J. Krause, F. Klemens, T. Henn, R. Trunk, H. Nirschl, Particle flow simulations with homogenised lattice Boltzmann methods, Particuology 34 (2017) 1–13, http://dx.doi.org/10.1016/j.partic.2016.11.001.

[30] R. Trunk, J. Marquardt, G. Thäter, H. Nirschl, M.J. Krause, Towards the simulation of arbitrarily shaped 3D particles using a homogenised lattice Boltzmann method, Comput. & Fluids 172 (2018) 621–631, http://dx.doi.org/10.1016/j.compfluid.2018.02.027.

[31] R. Trunk, T. Weckerle, N. Hafen, G. Thäter, H. Nirschl, M.J. Krause, Revisiting the homogenized lattice Boltzmann method with applications on particulate flows, Computation 9 (22) (2021) 11, http://dx.doi.org/10.3390/computation9020011.

[32] R. Trunk, C. Bretl, G. Thäter, H. Nirschl, M. Dorn, M.J. Krause, A study on shape-dependent settling of single particles with equal volume using surface resolved simulations, Computation 9 (44) (2021) 40, http://dx.doi.org/10.3390/computation9040040.

[33] N. Hafen, A. Dittler, M.J. Krause, Simulation of particulate matter structure detachment from surfaces of wall-flow filters applying lattice Boltzmann methods, Comput. & Fluids 239 (2022) 105381, http://dx.doi.org/10.1016/j.compfluid.2022.105381.

[34] N. Hafen, J.E. Marquardt, A. Dittler, M.J. Krause, Simulation of particulate matter structure detachment from surfaces of wall-flow filters for elevated velocities applying lattice Boltzmann methods, Fluids 8 (3) (2023) http://dx.doi.org/10.3390/fluids8030099.

[35] N. Hafen, J.E. Marquardt, A. Dittler, M.J. Krause, Simulation of dynamic rearrangement events in wall-flow filters applying lattice Boltzmann methods, Fluids 8 (77) (2023) 213, http://dx.doi.org/10.3390/fluids8070213.

[36] N. Hafen, J.E. Marquardt, A. Dittler, M.J. Krause, Simulation of dynamic rearrangement events in wall-flow filters applying lattice Boltzmann methods, Fluids 8 (7) (2023) 213, http://dx.doi.org/10.3390/fluids8070213.

[37] X. Li, F. Wang, D. Zhang, S. Gu, X. Gao, Fluid-solid interaction simulation for particles and walls of arbitrary polygonal shapes with a coupled lbm-imb-dem method, Powder Technol. 356 (2019) 177–192, http://dx.doi.org/10.1016/j.powtec.2019.08.006.

[38] J.E. Marquardt, U.J. Römer, H. Nirschl, M.J. Krause, A Discrete Contact Model for Complex Arbitrary-Shaped Convex Geometries, Particuology (2022) http://dx.doi.org/10.1016/j.partic.2022.12.005.

[39] S. Karimnejad, A.A. Delouei, H. Başağaoğlu, M. Nazari, M. Shahmardan, G. Falcucci, M. Lauricella, S. Succi, A review on contact and collision methods for multi-body hydrodynamic problems in complex flows, Commun. Comput. Phys. 32 (4) (2022) 899–950, http://dx.doi.org/10.4208/cicp.re-2022-0041.

[40] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E.M. Viggen, The Lattice Boltzmann Method, Graduate Texts in Physics, Springer, 2017, http://dx.doi.org/10.1007/978-3-319-44649-3.

[41] M.C. Sukop, D.T. Thorne, Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers, 2nd ed., Springer, 2006, http://dx.doi.org/10.1007/978-3-540-27982-2.

[42] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (3) (1954) 511–525, http://dx.doi.org/10.1103/PhysRev.94.511.

[43] Y.H. Qian, D. D'Humières, P. Lallemand, Lattice BGK models for Navier-Stokes equation, Europhys. Lett. (EPL) 17 (6) (1992) 479–484, http://dx.doi.org/10.1209/0295-5075/17/6/001.

[44] A. Kummerländer, T. Bingert, F. Bukreev, L.E. Czelusniak, D. Dapelo, N. Hafen, M. Heinzelmann, S. Ito, J. Jeßberger, H. Kusumaatmaja, J.E. Marquardt, M. Rennick, T. Pertzel, F. Prinz, M. Sadric, M. Schecher, S. Simonis, P. Sitter, D. Teutscher, M. Zhong, M.J. Krause, OpenLB release 1.7: Open source lattice Boltzmann code, 2024, http://dx.doi.org/10.5281/zenodo.10684609.

[45] M.J. Krause, A. Kummerländer, S.J. Avis, H. Kusumaatmaja, D. Dapelo, F. Klemens, M. Gaedtke, N. Hafen, A. Mink, R. Trunk, J.E. Marquardt, M.-L. Maier, M. Haussmann, S. Simonis, OpenLB—Open source lattice Boltzmann code, Comput. Math. Appl. 81 (2021) 258–288, http://dx.doi.org/10.1016/j.camwa.2020.04.033.

[46] A. Kupershtokh, D. Medvedev, D. Karpov, On equations of state in a lattice Boltzmann method, Comput. Math. Appl. 58 (2009) 965–974, http://dx.doi.org/10.1016/j.camwa.2009.02.024.

[47] B. Wen, C. Zhang, Y. Tu, C. Wang, H. Fang, Galilean invariant fluid–solid interfacial dynamics in lattice Boltzmann simulations, J. Comput. Phys. 266 (2014) 161–170, http://dx.doi.org/10.1016/j.jcp.2014.02.018.

[48] T. Henn, G. Thäter, W. Dörfler, H. Nirschl, M.J. Krause, Parallel dilute particulate flow simulations in the human nasal cavity, Comput. & Fluids 124 (2016) 197–207, http://dx.doi.org/10.1016/j.compfluid.2015.08.002.

[49] X. Yin, D.L. Koch, Hindered settling velocity and microstructure in suspensions of solid spheres with moderate Reynolds numbers, Phys. Fluids 19 (9) (2007) 093302, http://dx.doi.org/10.1063/1.2764109.

[50] L. Schiller, A. Neumann, Über die grundlegenden Berechnungen bei der Schwerkraftaufbereitung, Z. Vereines Deutscher Ingenieure 77 (1933) 318–321.

[51] S. Ergun, Fluid flow through packed columns, Chem. Eng. Progr. 48 (2) (1952) 89–94.

[52] I. Ginzbourg, D. d'Humières, Local second-order boundary methods for lattice Boltzmann models, J. Stat. Phys. 84 (5–6) (1996) 927–971, http://dx.doi.org/10.1007/bf02174124.

Jan E. Marquardt is a doctoral student at the Institute of Mechanical Process Engineering and Mechanics at the Karlsruhe Institute of Technology (KIT) and a contributor to the open-source C++ library OpenLB. He studied at the KIT in Germany, where he earned his B.Sc. (2017) and M.Sc (2019) degrees in Chemical and Process Engineering. His primary research interests are large-scale multiphysics simulations of particulate flows, with focus on particle-resolved CFD modeling and arbitrarily shaped particles.

Nicolas Hafen is a doctoral student at the Institute of Mechanical Process Engineering and Mechanics at the Karlsruhe Institute of Technology (KIT). He did both his Bachelors and Masters in Chemical and Process Engineering at the same university. His research revolves around fluid flow simulations in gas-filtering applications with main focus on fundamental investigations of particle-layer rearrangement event in ceramic wall-flow filters with Lattice Boltzmann Methods. In this context, he works on a unified framework for porous media and surface resolved particle modeling while being an active developer of the open-source C++ library OpenLB.

Mathias J. Krause studied mathematics with economics at the Universität Karlsruhe (TH) in Germany and the Cardiff University in Wales (2006). He completed his doctorate (2010) and habilitation (2020) at the Karlsruhe Institute of Technology (KIT). Leading the interdisciplinary Lattice Boltzmann Research Group (LBRG) since 2013, his research focuses on applied mathematics, applied computer science with focus on HPC, CFD and optimization under PDE constraints. He is initiator and author of OpenLB, an open-source software for solving large-scale biological, chemical, mechanical, and medical engineering problems. Recognized with multiple awards, he is a WIN-Kollegiat member at the Heidelberg Academy of Science.