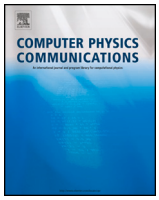
Computer Physics Communications 304 (2024) 109321

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



Computational Physics

A novel model for direct numerical simulation of suspension dynamics with arbitrarily shaped convex particles ☆

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

Keywords:
Discrete contact model
Particle-resolved simulation
Homogenized lattice Boltzmann method
Partially saturated methods
Hindered settling

ABSTRACT

This study presents an innovative direct numerical simulation approach for complex particle systems with irregular shapes and large numbers. Using partially saturated methods, it accurately models arbitrary shapes, albeit at considerable computational cost when integrating a compatible contact model. The introduction of a novel parallelization strategy significantly improves the performance of the contact model, enabling efficient four-way coupled simulations. Through hindered settling studies, the criticality of the explicit contact model for maintaining simulation accuracy is highlighted, especially at high particle volume fractions and low Archimedes numbers. The feasibility of simulating thousands of arbitrarily shaped convex particles is demonstrated with up to 1934 surface-resolved particles. The study also confirms the grid independence and linear convergence of the method. It shows for the first time that cube swarms settle 13 to 26% slower than swarms of volume-equivalent spheres across different Archimedes numbers (500 to 2000) and particle volume fractions (10 to 30%). These findings emphasize the shape dependence of particle systems and suggest avenues for exploring their nuanced dynamics.

1. Introduction

Understanding the dynamics of suspensions, which consist of solid particles dispersed in a liquid [1], 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 [2–9] and clustering [10–12], 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 [13], rod-like particles [14], sand grains [15], and fibers [16]. 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 [17]. 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 [18,19].

☆ The review of this paper was arranged by Prof. Andrew Hazel.

* Corresponding author at: Lattice Boltzmann Research Group, Karlsruhe Institute of Technology, Straße am Forum 8, Karlsruhe, 76131, Germany.
E-mail address: jan.marquardt@kit.edu (J.E. Marquardt).

https://doi.org/10.1016/j.cpc.2024.109321
Received 4 January 2024; Received in revised form 3 July 2024; Accepted 15 July 2024
Available online 18 July 2024

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 [20–22] or level-set DEM [23–25]. 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 [26,27], which leads to inaccuracies. Another approach omits the back-coupling to the fluid [28], which, however, is essential especially at high particle volume fractions [17].

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 [29]. 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 [30]. 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 [31]. Its derivatives were used extensively in the investigation of particle flows [32,33]. A notable derivative of PSM is the homogenized lattice Boltzmann method (HLBM) introduced by Krause et al. [34], which has been successfully used to study different particle shapes [35–39]. Particularly noteworthy are studies on the shape-dependent sedimentation behavior of single particles [40]. With the recent development of a compatible contact model [41], 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 [42], 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 [43–45], 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 2 presents the models employed to describe the fluid and particle behavior, while Section 3 discusses the numerical techniques utilized to solve the model system. In Section 4, we outline the proposed parallelization strategy, which is then applied to investigate hindered settling in Section 5. Lastly, Section 6 summarizes the key findings and conclusions drawn from this study.

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 [41,42]. The following sections give a brief overview of these models.

2.1. Fluid

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

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

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.

2.2. Particle

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

$$m_\mathrm{p} \frac{\partial \boldsymbol{u}_\mathrm{p}}{\partial t} = \boldsymbol{F}_\mathrm{p}, \tag{2}$$

while rotation is described by

$$\boldsymbol{I}_\mathrm{p} \frac{\partial \boldsymbol{\omega}_\mathrm{p}}{\partial t} + \boldsymbol{\omega}_\mathrm{p} \times (\boldsymbol{I}_\mathrm{p} \cdot \boldsymbol{\omega}_\mathrm{p}) = \boldsymbol{T}_\mathrm{p}. \tag{3}$$

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.

2.3. Contact

To account for interactions of even complex geometries, we employ a model introduced by Nassauer and Kuna [46]. Here, the normal contact force is given by

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

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

$$E^* = \left(\frac{1 - \nu_\mathrm{A}^2}{E_\mathrm{A}} + \frac{1 - \nu_\mathrm{B}^2}{E_\mathrm{B}} \right)^{-1}, \tag{5}$$

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 [47] and use

$$c = \begin{cases} 1.5 \frac{(1-e)(11-e)}{(1+9e)u_0} & \text{for } u_0 > 0 \\ 0 & \text{for } u_0 \leq 0 \end{cases}, \tag{6}$$

where u_0 is the magnitude of the relative velocity at the initial contact [37].

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 [46] describe the tangential force using the equation

$$\boldsymbol{F}_{c,t} = -\frac{\boldsymbol{u}_{AB,t}(\boldsymbol{x}_c)}{||\boldsymbol{u}_{AB,t}(\boldsymbol{x}_c)||}\left(\left(2\mu_s^* - \mu_k\right)\frac{a^2}{a^4+1} + \mu_k - \frac{\mu_k}{a^2+1}\right)||\boldsymbol{F}_{c,n}||, \tag{7}$$

with

$$\mu_s^* = \mu_s\left(1 - 0.09\left(\frac{\mu_k}{\mu_s}\right)^4\right), \tag{8}$$

and

$$a = \frac{||\boldsymbol{u}_{AB,t}(\boldsymbol{x}_c)||}{u_k}. \tag{9}$$

$\boldsymbol{u}_{AB,t}$ represents the relative tangential velocity at the contact point $\boldsymbol{x}_c$ 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 $u_k = 0.001\text{m/s}$.

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 [48].

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 3.1 and 3.2 are given in lattice units.

3.1. Lattice Boltzmann method

To solve the incompressible Navier–Stokes equations, the lattice Boltzmann method (LBM) [49–51] 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 [49,50]. 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

$$f_i^*(\boldsymbol{x},t) = f_i(\boldsymbol{x},t) + \Omega_i(\boldsymbol{x},t) + S_i(\boldsymbol{x},t), \tag{10}$$

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

$$f_i(\boldsymbol{x}+\boldsymbol{c}_i, t+1) = f_i^*(\boldsymbol{x},t), \tag{11}$$

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 [52]

$$\Omega_i(\boldsymbol{x},t) = -\frac{1}{\tau}(f_i(\boldsymbol{x},t) - f_i^{eq}(\rho_f, \boldsymbol{u}_f)), \tag{12}$$

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

$$f_i^{eq}(\rho_f, \boldsymbol{u}_f) = w_i\rho_f\left(1 + \frac{\boldsymbol{c}_i\cdot\boldsymbol{u}_f}{c_s^2} + \frac{(\boldsymbol{c}_i\cdot\boldsymbol{u}_f)^2}{2c_s^4} - \frac{\boldsymbol{u}_f\cdot\boldsymbol{u}_f}{2c_s^2}\right), \tag{13}$$

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 $\rho_f(\boldsymbol{x},t) = \sum_i f_i(\boldsymbol{x},t)$ and momentum $\rho_f\boldsymbol{u}_f(\boldsymbol{x},t) = \sum_i \boldsymbol{c}_i f_i(\boldsymbol{x},t)$.

All studies in this paper were performed using the LBM implemented in the open source software OpenLB [53,54].

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. [34]. 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 [32,40,55]. The signed distance to the particle surface is used as a parameter to integrate any shape in this level set function [42].

To account for the influence of the particles on the fluid, we use an exact difference method (EDM) introduced by Kupershtokh et al. [56], as its adaptation has been reported to be superior [40]. It introduces the following source term in Eq. (10)

$$S_i(\boldsymbol{x},t) = f_i^{eq}(\rho_f, \boldsymbol{u}_f + \Delta\boldsymbol{u}_f) - f_i^{eq}(\rho_f, \boldsymbol{u}_f). \tag{14}$$

The required velocity difference $\Delta\boldsymbol{u}_f(\boldsymbol{x},t)$ is computed by a convex combination of the fluid and particle velocities [40]

$$\Delta\boldsymbol{u}_f(\boldsymbol{x},t) = (B(\boldsymbol{x},t) - 1)\left(\boldsymbol{u}_p(\boldsymbol{x},t) - \boldsymbol{u}_f(\boldsymbol{x},t)\right). \tag{15}$$

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. [57]. The local force is then

$$\boldsymbol{F}_h(\boldsymbol{x},t) = \sum_i (\boldsymbol{c}_i - \boldsymbol{u}_p(\boldsymbol{x},t))f_i(\boldsymbol{x}+\boldsymbol{c}_i,t) + (\boldsymbol{c}_i + \boldsymbol{u}_p(\boldsymbol{x},t))f_{\bar{i}}(\boldsymbol{x},t). \tag{16}$$

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

$$\boldsymbol{F}_p(t) = \sum_{\boldsymbol{x}_b} \boldsymbol{F}_h(\boldsymbol{x}_b, t). \tag{17}$$

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

$$\boldsymbol{T}_p(t) = \sum_{\boldsymbol{x}_b} (\boldsymbol{x}_b - \boldsymbol{X}_p) \times \boldsymbol{F}_h(\boldsymbol{x}_b, t). \tag{18}$$

While $\boldsymbol{x}_b$ resides within the particle, the formulation in Eq. (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.

3.3. Discrete contacts

The original discrete contact model [41] 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 3.3.1, while Section 3.3.2 introduces a method for considering periodic boundaries.

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 ($d_s < \sqrt{0.75}\Delta x$), 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 [41].

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 [40,58]. 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.

4. Parallelization strategy

Given the computational expense associated with the bounding box correction and contact force calculation introduced in Section 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 [53], and in the public repository.[1]

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

4.1. Definitions

For clarity and consistency, we provide the following definitions that will be utilized throughout the sections below [42]:

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.

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 [49,59]. 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.

Marquardt et al. [42] 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.

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 [59].

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 Fig. 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,

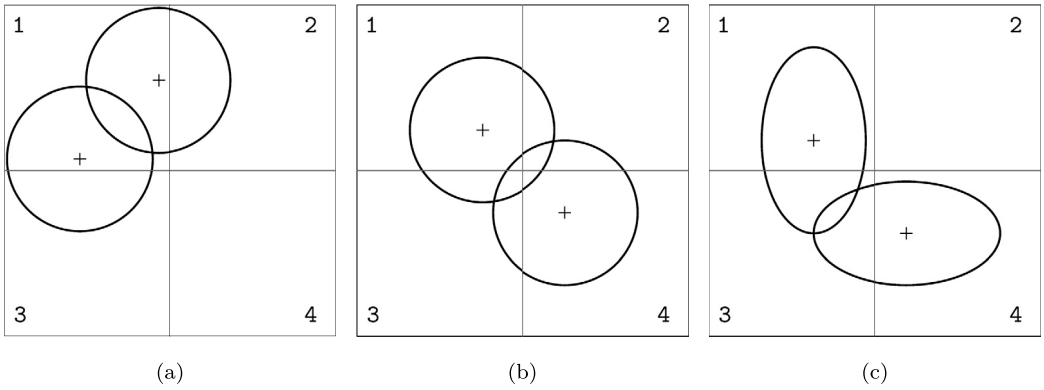


Fig. 1. Illustration of three types of particle-particle contacts: The centers of mass of both particles are in the same block (a), the centers of mass of both particles are in different blocks, but both particles are known to at least one of the responsible PUs (b), and none of the responsible PUs knows both particles, only a third PU knows all data (c).

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 the surface intersects the corresponding block. As the particle decomposition scheme [42] aims to avoid unnecessary complexity, however, the actual algorithm does not check for intersections and communicates the particle data to all neighbors instead.

In Fig. 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 Fig. 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 Fig. 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.

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 Fig. 2.

4.3.2. Communication of the detected contacts

Rough on-lattice contact detection, as described in [41], 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 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 Fig. 3, and includes critical details such as global particle IDs or global particle

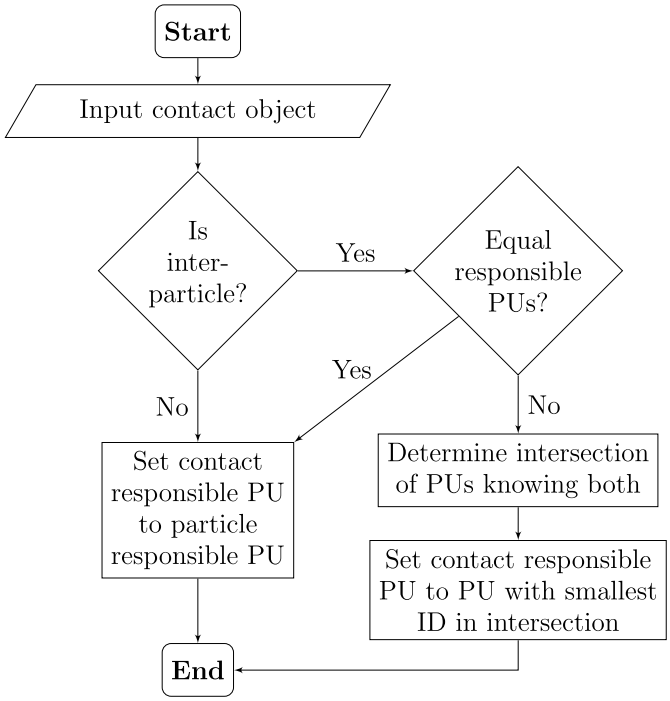


Fig. 2. Visualization of the algorithm to determine a contact responsible PU.

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.

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.

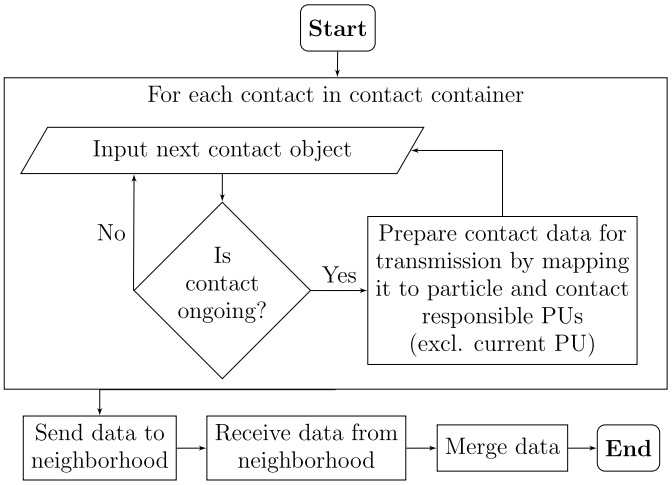


Fig. 3. Basic procedure for communicating detected contacts. Note that this step takes place after the determination of the contact responsible PU (cf. Section 4.3.1), during which the destination PUs for this step may be determined to save computational resources.

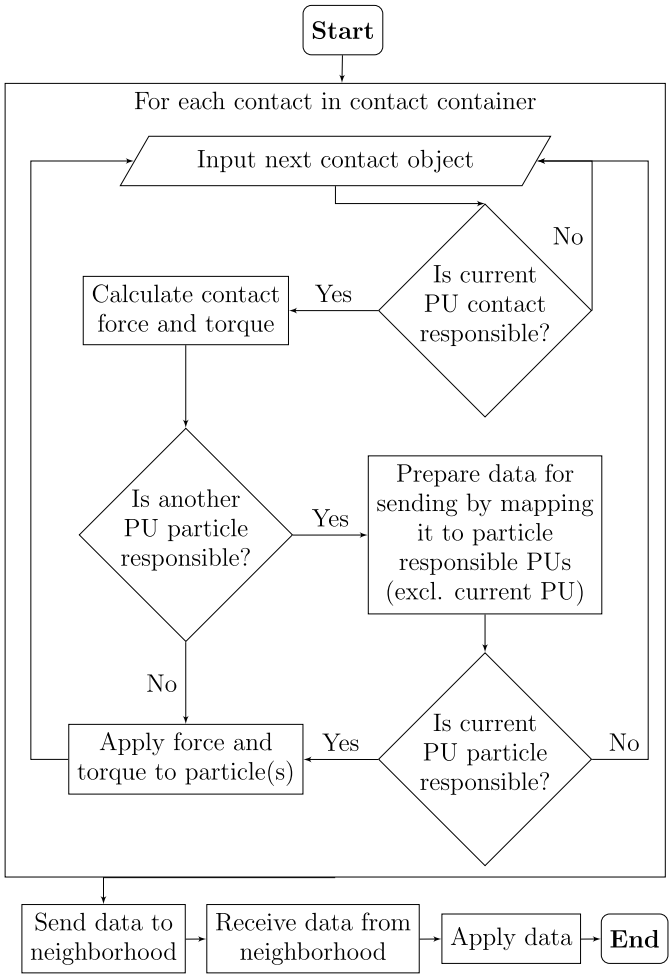


Fig. 4. Visualization of the steps required to communicate the results of the contact treatment.

To reduce the computational effort, this step is combined with the contact force and torque calculation as shown in Fig. 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.

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. Fig. 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. (6).

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

4.4. *Time step algorithm*

To give a structured overview, Algorithm 1 outlines the basic procedure. It is worth noting that it includes sub time steps with constant hydrodynamic forces [41]. 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 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 [42]
    for all sub time steps do
        Compute contact forces;                         ▷ See [41]
        Communicate contact forces;                     ▷ See 4.3.3
        Apply external forces;                          ▷ Such as gravity
        Solve equations of motion;
        Communicate post-treatment contacts;            ▷ See 4.3.4
        Evaluate particle assignment;                   ▷ See [42]
        Communicate data and assignment;                ▷ See [42]
        Couple particles to fluid with contact detection; ▷ See [41]
        Determine responsible PUs;                      ▷ See 4.3.1
        Communicate detected contacts;                  ▷ See 4.3.2
        Apply periodic boundary to contacts (optional); ▷ See 3.3.2
    end
    Perform collision and streaming;
    Increase time step;
end
```

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.

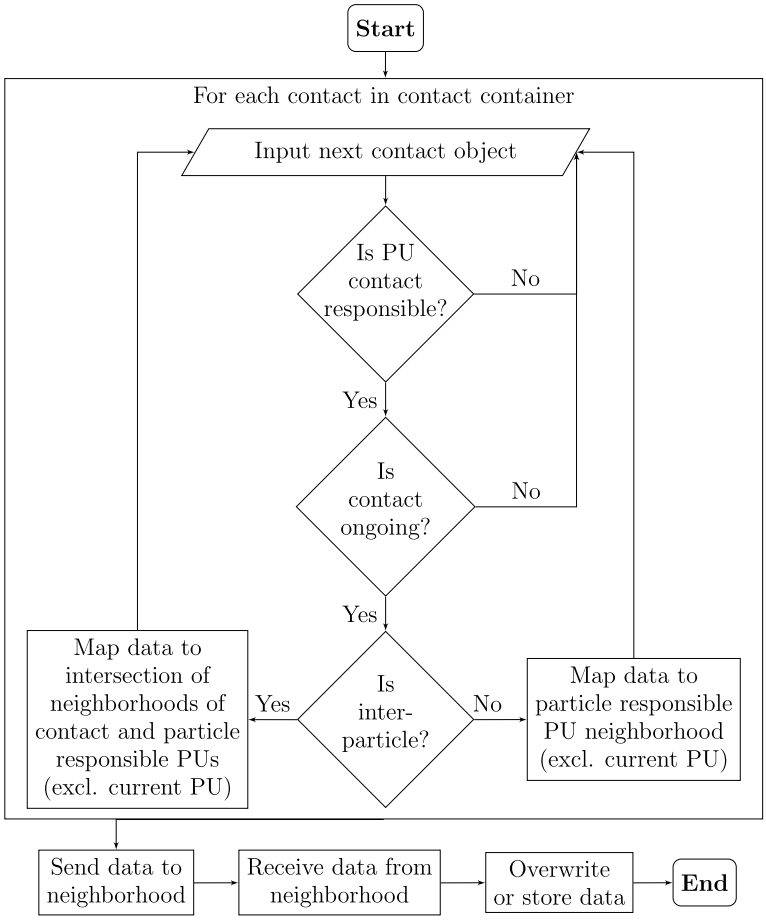


Fig. 5. Illustration of the communication of contacts after the contact treatment.

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 [60]

$$\frac{\bar{u}_\mathrm{p}}{u^*} = k(1-\phi_\mathrm{p})^n. \tag{19}$$

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

$$u^* = \sqrt{\frac{4gD_\mathrm{s}}{3C_\mathrm{d}}\left(\frac{\rho_\mathrm{p}-\rho_\mathrm{f}}{\rho_\mathrm{f}}\right)}, \tag{20}$$

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 [61]

$$C_\mathrm{d} = \frac{24}{Re}\left(1+0.15Re^{0.687}\right), \tag{21}$$

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 [3] propose

$$n = \begin{cases} 4.65 & \text{for } Re < 0.2 \\ 4.35Re^{-0.03} & \text{for } 0.2 \le Re < 1 \\ 4.45Re^{-0.1} & \text{for } 1 \le Re < 500 \\ 2.39 & \text{for } 500 \le Re \end{cases}. \tag{22}$$

Using a power law-based approach, Garside and Al-Dibouni [6] suggest

$$n = \frac{5.1+0.27Re^{0.9}}{1+0.1Re^{0.9}}. \tag{23}$$

For the latter correlation, superior accuracy has been reported [60]. Note that Richardson and Zaki [3] use a Reynolds number that depends on the velocity of a single particle in an infinite fluid, while Garside and Al-Dibouni [6] 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 [7,8,13] suggest including a prefactor $k \in [0.8, 0.9]$. Yao et al. [12] 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

$$k = 0.89\exp\left(-\frac{Ar}{10^5}\right), \tag{24}$$

where the Archimedes number is defined as

$$Ar = \frac{gD_\mathrm{s}^3\rho_\mathrm{f}(\rho_\mathrm{p}-\rho_\mathrm{f})}{\eta^2}. \tag{25}$$

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 Fig. 6.

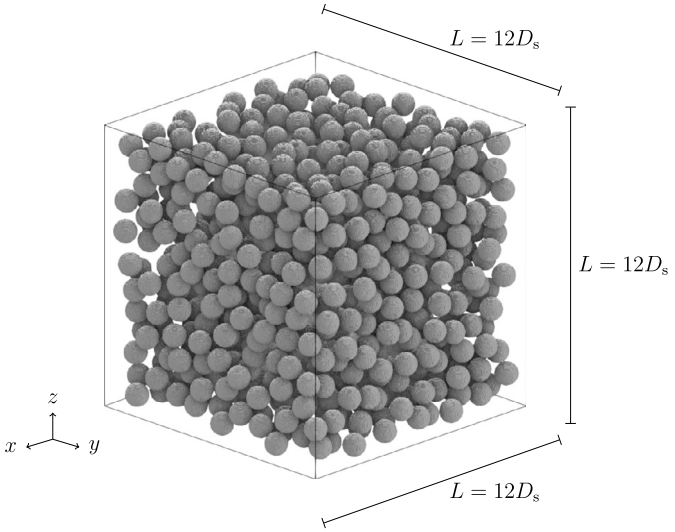


Fig. 6. Simulation domain with spherical particles on their preassigned initial positions using a resolution of $N = 27$ cells per sphere diameter and particle volume fraction of about 30%.

Initially, the particles rest, but they experience an acceleration in the z-direction due to the force $F_g = (\rho_f - \rho_p)V_p g$, where V_p represents the volume of a single particle. Following previous studies [9,11,60,62], 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 [12]. 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 [41], 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 $\mu_k = 0.16$ [63]. The coefficient of static friction is set to 0.32. Similar to studies in literature [11,62], 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 $\tau > 0.5$ [49], 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 $\tau < 0.55$ [49]. Fortunately, the findings presented in Section 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 simulations using $L = 12D_s$ and $N = 18$ shown in Fig. 7 illustrate that this is sufficient. Fig. 7 shows the normalized average settling velocity $\bar{u}_p/u^*$ 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 $\phi_p \in \{0.1, 0.3\}$ 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.

5.2. 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 Fig. 8. This figure shows the relative error, calculated using the L^2 norm as described in [49], 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 crosses. The figure also includes lines representing the experimental orders of convergence (EOC) with values of both 1 and 2.

In Fig. 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.

5.3. 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. Fig. 9 shows the explicit parameters of the different setups and plots the

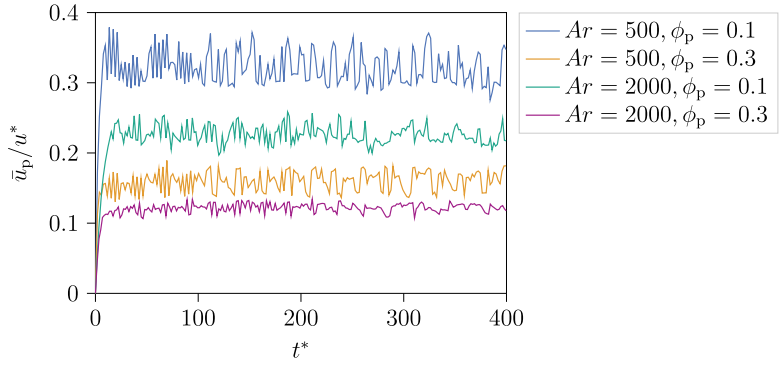


Fig. 7. Normalized average settling velocity versus the normalized time t^*. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

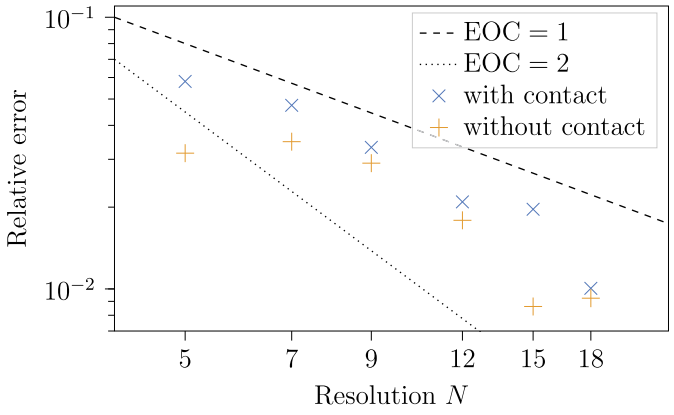


Fig. 8. Relative error with and without the contact model in L^2 norm versus the resolution of the sphere's diameter N. The references use the resolution $N = 27$.

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 Fig. 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

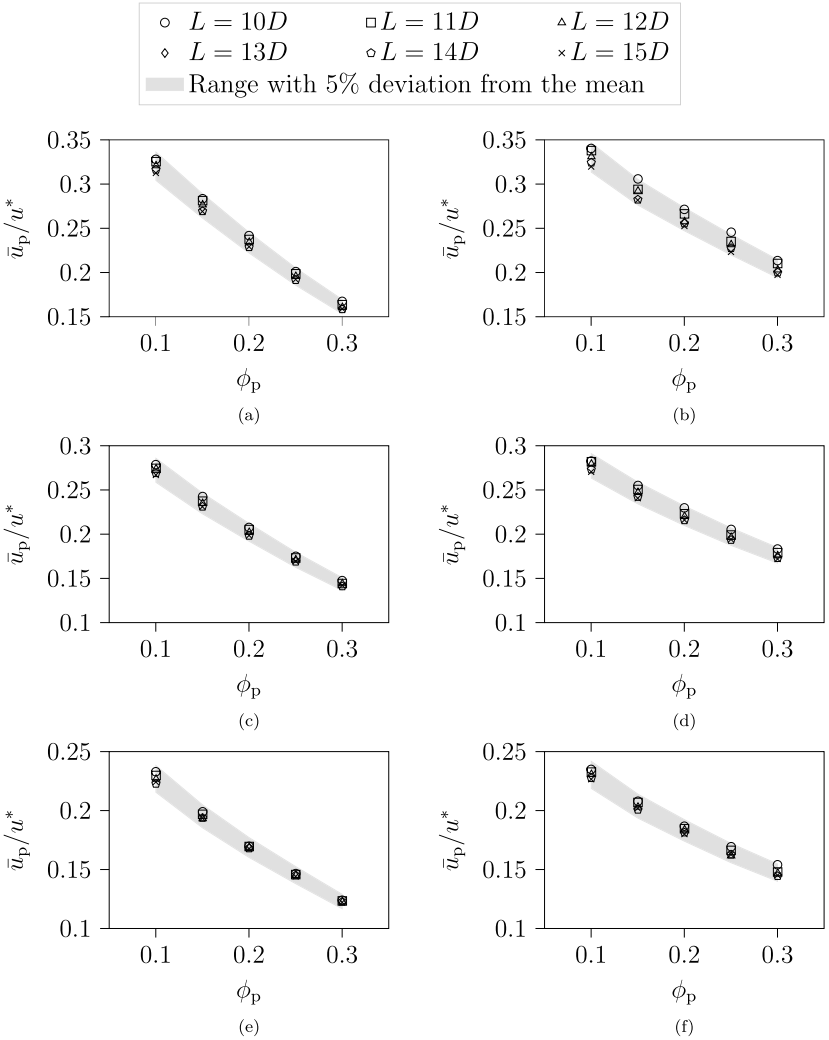


Fig. 9. Visualization of the domain dependence of the results by plotting the normalized mean sedimentation velocity versus the particle volume fraction ϕ_p for different domain sizes. On the left, the simulations were performed with an explicit contact model and on the right without. The top plots consider $Ar = 500$, the middle plots $Ar = 1000$, and the bottom plots $Ar = 2000$.

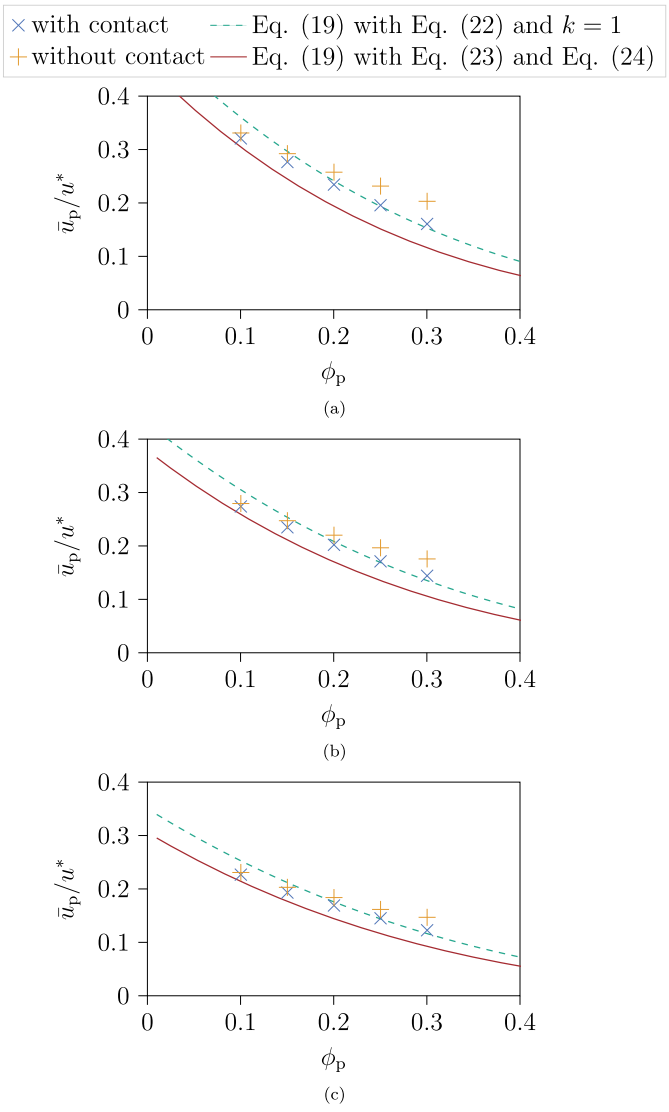


Fig. 10. Comparison of simulation results with and without the use of an explicit contact model with correlations from literature. (a) $Ar = 500$, (b) $Ar = 1000$, and (c) $Ar = 2000$.

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 Fig. 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 [3], i.e. Eq. (19) with Eq. (22) and $k = 1$ as a dashed green line. The red solid line shows the results for the correlation of Garside and Al-Dibouni [6] using the prefactor k as suggested by Yao et al. [12], i.e. Eq. (19) with Eq. (23) and Eq. (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. Eq. (19) with Eq. (23) and Eq. (24) (solid red line), however, predicts lower velocities than Eq. (19) with Eq. (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 [3], 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 [40].

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 due to its limitations, described in Section 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.

Fig. 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 $\phi_p \approx 0.3$ 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 $\phi_p \approx 0.1$.

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 Fig. 11(a), the impact is quite significant, as the MLUPs can decrease by a factor of about 3. However, increasing the resolution, see Fig. 11(b), or the domain size, see Fig. 11(c), decreases the impact of the contact treatment. It is almost negligible for large problem sizes, especially for small particle volume fractions, see Fig. 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

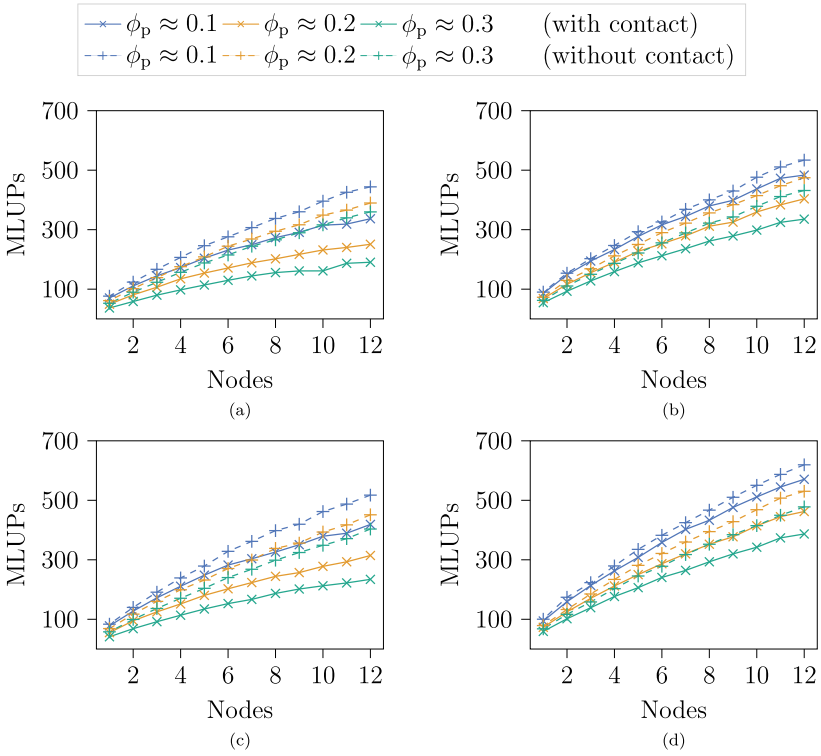


Fig. 11. Comparison of the MLUPs versus the number of nodes used for simulations with and without explicit contact treatment for different particle volume fractions. At the top, the simulations with an edge length of $L = 12D_s$ are shown. At the bottom, the edge length is $L = 15D_s$. On the left, a resolution of $N = 18$ and on the right of $N = 27$ is used.

now feasible to consider thousands of surface resolved particles thanks to the proposed parallelization strategy.

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 5.1. Note that the initial positions of the particles are also the same, as shown for a single parameter set in Fig. 6 and Fig. 12.

Fig. 13 visualizes the fluid velocity field and particles for $Ar = 2000$ and $\phi_p \approx 0.3$ 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 1 compares the hindered settling behavior of a collective of spheres and cubes by displaying the average settling velocity of the spheres $\bar{u}_{p,spheres}$ and cubes $\bar{u}_{p,cubes}$ 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^* to ensure comparability.

Table 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.

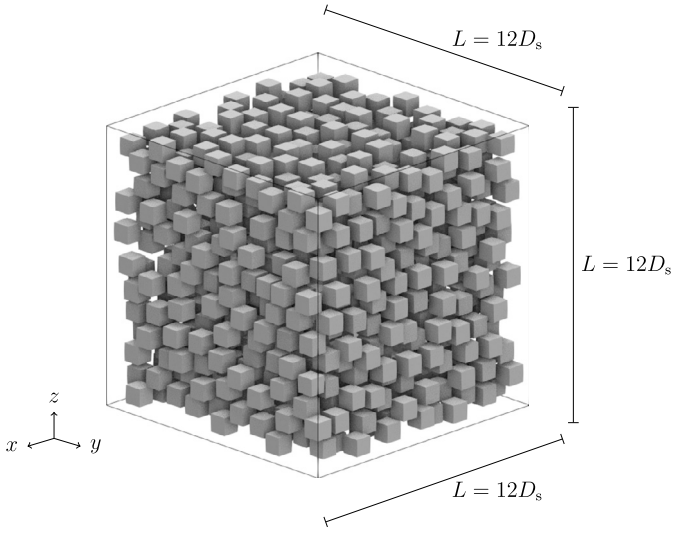


Fig. 12. Simulation domain with 991 cubic particles at their preassigned initial positions using a resolution of $N = 27$ cells per volume-equivalent sphere diameter and particle volume fraction $\phi_p \approx 0.3$.

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

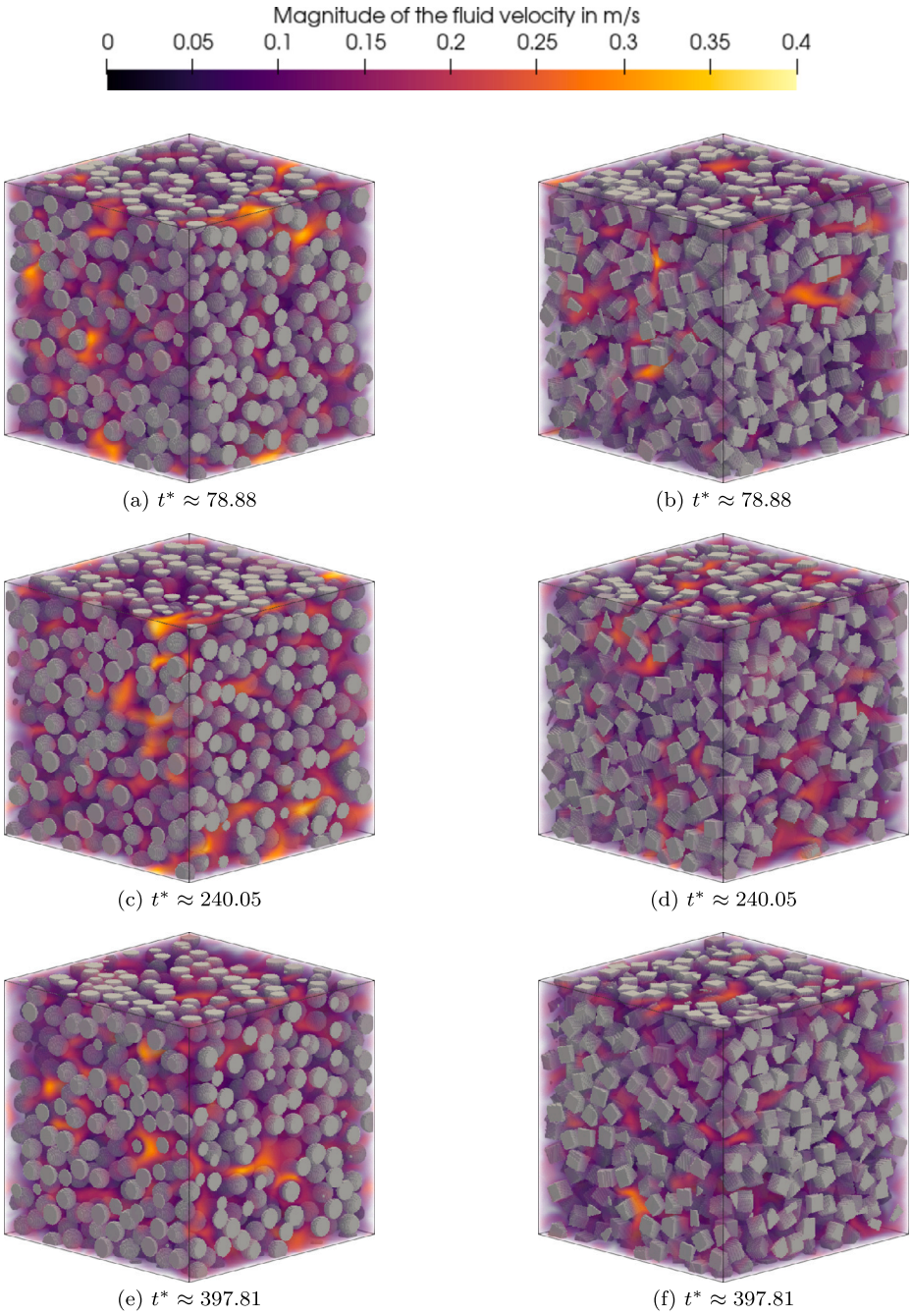


Fig. 13. Velocity field around the particles at different normalized times t^* for $Ar = 2000$ and $\phi_p \approx 0.3$.

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 [13], rod-like particles [14], sand grains [15], and fibers [16]. This leads to the assumption that non-spherical particles generally experience greater hindered settling effects.

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. [41], 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

Table 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

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.

Nomenclature

Acronyms

BGK	Bhatnagar–Gross–Krook
DEM	discrete element method
DNS	direct numerical simulation
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
PU	processing unit

Roman symbols

Ar	Archimedes number
B	weighting factor
C_d	drag coefficient
$\boldsymbol{c}$	discrete velocity
c	damping factor
c_s	lattice speed of sound
D	diameter
d	indentation depth
$\dot{d}$	temporal change of indentation depth
d_s	signed distance
E	Young's modulus
E^*	effective Young's modulus
e	coefficient of restitution
$\boldsymbol{F}$	force
F_g	combination of weight and buoyancy
f	particle population
f^*	post-collision particle population
g	standard gravity
$\boldsymbol{I}$	moment of inertia
k	prefactor in hindered settling correlations
L	edge length of periodic simulation domain
m	mass
N	resolution
n	expansion index
$\boldsymbol{n}_c$	contact normal
p	pressure
Re	Reynolds number
S	source term
$\boldsymbol{T}$	torque
t	time
$\boldsymbol{u}$	velocity
u^*	reference settling velocity of a single sphere
u_0	initial relative velocity of two objects in contact
u_k	velocity for transition from static to kinetic friction
V	volume
w	weight for the equilibrium distribution calculation
$\boldsymbol{X}$	center of mass
$\boldsymbol{x}$	position

Greek symbols

Δt	time step size
Δx	grid spacing
η	dynamic viscosity
μ_k	coefficient of kinetic friction
μ_s	coefficient of static friction
ν	Poisson's ratio
ρ	density
τ	relaxation time
ϕ	volume fraction
Ω	collision operator
$\boldsymbol{\omega}$	angular velocity

Subscripts

A	refers to an object A
B	refers to an object B
b	refers to positions inside a particle's boundary
c	refers to a particle-particle or particle-wall interaction
f	refers to the fluid
h	refers to the hydrodynamic force
i	refers to the corresponding discrete velocity
n	refers to the normal direction
p	refers to the particle's center of mass
s	refers to a sphere
t	refers to the tangential direction

CRediT authorship contribution statement

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

Software, Methodology, Conceptualization. **Mathias J. Krause:** Writing – review & editing, Supervision, Software, Resources, Project administration, Funding acquisition.

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.

Acknowledgements

This research was funded by the DFG (German Research Foundation) under 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 HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

References

[1] S. Slomkowski, J.V. Alemán, R.G. Gilbert, M. Hess, K. Horie, R.G. Jones, P. Kubisa, I. Meisel, W. Mormann, S. Penczek, R.F.T. Stepto, Terminology of polymers and polymerization processes in dispersed systems (iupac recommendations 2011), Pure Appl. Chem. 83 (12) (2011) 2229–2259, https://doi.org/10.1351/pac-rec-10-06-03.

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

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

[4] D.R. Oliver, The sedimentation of suspensions of closely-sized spherical particles, Chem. Eng. Sci. 15 (3) (1961) 230–242, https://doi.org/10.1016/0009-2509(61)85026-4.

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

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

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

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

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

[10] M. Uhlmann, T. Doychev, Sedimentation of a dilute suspension of rigid spheres at intermediate Galileo numbers: the effect of clustering upon the particle motion, J. Fluid Mech. 752 (2014) 310–348, https://doi.org/10.1017/jfm.2014.330.

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

[12] 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, https://doi.org/10.1017/jfm.2021.470.

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

[14] M.A. Turney, M.K. Cheung, R.L. Powell, M.J. McCarthy, Hindered settling of rod-like particles measured with magnetic resonance imaging, AIChE J. 41 (2) (1995) 251–257, https://doi.org/10.1002/aic.690410207.

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

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

[17] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, B. Van Wachem, Computational Fluid Dynamics for Engineers, Cambridge University Press, 2011.

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

[19] H. Kruggel-Emden, S. Rickelt, S. Wirtz, V. Scherer, A study on the validity of the multi-sphere discrete element method, Powder Technol. 188 (2) (2008) 153–165, https://doi.org/10.1016/j.powtec.2008.04.037.

[20] 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, https://doi.org/10.1007/s40571-018-0198-3.

[21] A.D. Rakotonirina, A. Wachs, Grains3D, a flexible DEM approach for particles of arbitrary convex shape—Part II: parallel implementation and scalable performance, Powder Technol. 324 (2018) 18–35, https://doi.org/10.1016/j.powtec.2017.10.033.

[22] A. Wachs, L. Girolami, G. Vinay, G. Ferrer, Grains3D, a flexible DEM approach for particles of arbitrary convex shape—Part I: numerical model and validations, Powder Technol. 224 (2012) 374–389, https://doi.org/10.1016/j.powtec.2012.03.023.

[23] R. Kawamoto, E. Andò, G. Viggiani, J.E. Andrade, Level set discrete element method for three-dimensional computations with triaxial case study, J. Mech. Phys. Solids 91 (2016) 1–13, https://doi.org/10.1016/j.jmps.2016.02.021.

[24] D.L.H. van der Haven, I.S. Fragkopoulos, J.A. Elliott, A physically consistent discrete element method for arbitrary shapes using volume-interacting level sets, Comput. Methods Appl. Mech. Eng. 414 (2023) 116165, https://doi.org/10.1016/j.cma.2023.116165.

[25] S. Wang, D. Liang, S. Ji, Dem study on mixing behaviors of concave-shaped particles in rotating drum based on level-set method, Powder Technol. 430 (2023) 118961, https://doi.org/10.1016/j.powtec.2023.118961.

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

[27] 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, https://doi.org/10.1016/j.cageo.2016.01.011.

[28] 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, https://doi.org/10.3390/min12040480.

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

[30] T. Nagata, M. Hosaka, S. Takahashi, K. Shimizu, K. Fukuda, S. Obayashi, A simple collision algorithm for arbitrarily shaped objects in particle-resolved flow simulation using an immersed boundary method, Int. J. Numer. Methods Fluids 92 (10) (2020) 1256–1273, https://doi.org/10.1002/fld.4826.

[31] D.R. Noble, J.R. Torczynski, A lattice-Boltzmann method for partially saturated computational cells, Int. J. Mod. Phys. C 09 (08) (1998) 1189–1201, https://doi.org/10.1142/S0129183198001084.

[32] 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, https://doi.org/10.1016/j.camwa.2020.04.022.

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

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

[35] 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, https://doi.org/10.1016/j.compfluid.2022.105381.

[36] 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), https://doi.org/10.3390/fluids8030099.

[37] 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, https://doi.org/10.3390/fluids8070213.

[38] 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, https://doi.org/10.1016/j.compfluid.2018.02.027.

[39] 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, https://doi.org/10.3390/computation9040040.

[40] 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, https://doi.org/10.3390/computation9020011.

[41] J.E. Marquardt, U.J. Römer, H. Nirschl, M.J. Krause, A discrete contact model for complex arbitrary-shaped convex geometries, Particuology 80 (2023) 180–191, https://doi.org/10.1016/j.partic.2022.12.005.

[42] J.E. Marquardt, N. Hafen, M.J. Krause, A novel particle decomposition scheme to improve parallel performance of fully resolved particulate flow simulations, J. Comput. Sci. 78 (2024) 102263, https://doi.org/10.1016/j.jocs.2024.102263.

[43] B. Blais, L. Barbeau, V. Bibeau, S. Gauvin, T.E. Geitani, S. Golshan, R. Kamble, G. Mirakhori, J. Chaouki, Lethe: an open-source parallel high-order adaptative CFD

solver for incompressible flows, SoftwareX 12 (2020) 100579, https://doi.org/10.1016/j.softx.2020.100579.
[44] T. El Geitani, S. Golshan, B. Blais, Toward high-order CFD-DEM: development and validation, Ind. Eng. Chem. Res. 62 (2) (2023) 1141–1159, https://doi.org/10.1021/acs.iecr.2c03546.
[45] S. Golshan, P. Munch, R. Gassmöller, M. Kronbichler, B. Blais, Lethe-DEM: an open-source parallel discrete element solver with load balancing, Comput. Part. Mech. 10 (1) (2022) 77–96, https://doi.org/10.1007/s40571-022-00478-6.
[46] B. Nassauer, M. Kuna, Contact forces of polyhedral particles in discrete element method, Granul. Matter 15 (3) (2013) 349–355, https://doi.org/10.1007/s10035-013-0417-9.
[47] A.S. Carvalho, J.M. Martins, Exact restitution and generalizations for the Hunt–Crossley contact model, Mech. Mach. Theory 139 (2019) 174–194, https://doi.org/10.1016/j.mechmachtheory.2019.03.028.
[48] P. Flores, Contact mechanics for dynamical systems: a comprehensive review, Multibody Syst. Dyn. (Oct. 2021), https://doi.org/10.1007/s11044-021-09803-y.
[49] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E.M. Viggen, The Lattice Boltzmann Method, Graduate Texts in Physics, Springer, 2017.
[50] S. Succi, The Lattice Boltzmann Equation: for Fluid Dynamics and Beyond, Oxford University Press, 2001.
[51] M.C. Sukop, D.T. Thorne, Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers, 2nd edition, Springer, 2006.
[52] 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, https://doi.org/10.1103/PhysRev.94.511.
[53] 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, https://dx.doi.org/10.5281/zenodo.10684609, Feb. 2024.
[54] 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, https://doi.org/10.1016/j.camwa.2020.04.033.
[55] N. Hafen, J.R. Thieringer, J. Meyer, M.J. Krause, A. Dittler, Numerical investigation of detachment and transport of particulate structures in wall-flow filters using lattice Boltzmann methods, J. Fluid Mech. 956 (2023) A30, https://doi.org/10.1017/jfm.2023.35.
[56] A. Kupershtokh, D. Medvedev, D. Karpov, On equations of state in a lattice Boltzmann method, Comput. Math. Appl. 58 (2009) 965–974, https://doi.org/10.1016/j.camwa.2009.02.024.
[57] 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, https://doi.org/10.1016/j.jcp.2014.02.018.
[58] Y. Chen, G. Jin, P. Zhang, S.A. Galindo-Torres, A. Scheuermann, L. Li, An efficient framework for particle-fluid interaction using discrete element lattice Boltzmann method: coupling scheme and periodic boundary condition, Comput. Fluids 208 (2020) 104613, https://doi.org/10.1016/j.compfluid.2020.104613.
[59] 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, https://doi.org/10.1016/j.compfluid.2015.08.002.
[60] 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, https://doi.org/10.1063/1.2764109.
[61] L. Schiller, A. Neumann, Über die grundlegenden Berechnungen bei der Schwerkraftaufbereitung, Z. Ver. Dtsch. Ing. 77 (1933) 318–321.
[62] D.P. Willen, A.J. Sierakowski, G. Zhou, A. Prosperetti, Continuity waves in resolved-particle simulations of fluidized beds, Phys. Rev. Fluids 2 (11) (2017) 114305, https://doi.org/10.1103/PhysRevFluids.2.114305.
[63] H. Tang, R. Song, Y. Dong, X. Song, Measurement of restitution and friction coefficients for granular particles and discrete element simulation for the tests of glass beads, Materials 12 (19) (2019) 3170, https://doi.org/10.3390/ma12193170.