A hybrid parallel approach for fully resolved simulations of particle-laden flows in sediment transport

Zhengping Zhu\textsuperscript{a,c}, Ruifeng Hu\textsuperscript{b}, Lian Shen\textsuperscript{c}, Xiaojing Zheng\textsuperscript{a,b,*}

\textsuperscript{a}Research Center for Applied Mechanics, School of Mechano-Electronic Engineering, Xidian University, Xi’an 710071, China
\textsuperscript{b}Center for Particle-Laden Turbulence, Key Laboratory of Mechanics on Disaster and Environment in Western China, Ministry of Education, and College of Civil Engineering and Mechanics, Lanzhou University, Lanzhou 730000, China
\textsuperscript{c}St. Anthony Falls Laboratory and Department of Mechanical Engineering, University of Minnesota, Minneapolis, Minnesota 55455, USA

Abstract

A hybrid parallel approach for fully resolved simulations of particle-laden flows in sediment transport is proposed. To overcome the challenges of load imbalance in the traditional domain decomposition method when encountering highly uneven distributions of particles in space, we develop a hybrid parallel approach adopting the domain decomposition method for the carrier phase and a mirror domain technique for the disperse phase. We modify the mirror domain technique originally developed for point particles to fully resolved particle simulations, which are more challenging since a finite-sized particle may be split into different subdomains; thus, more complex treatments of particle-fluid interactions are needed. By utilizing the mirror domain technique, in which each processor handles nearly the same number of particles regardless of the particle spatial distribution, excellent load balance is achieved. The present hybrid parallel approach also shows strong scalability and high parallel efficiency in a test of a fully resolved simulation case of sediment transport. Furthermore, a novel memory optimization method is proposed for spherical particles of equal size, which can substantially reduce the memory cost and enable the simulation of millions of fully resolved particles on a common highly parallel computing platform. Our code is validated by several benchmark cases, and the results show good agreement with

\textsuperscript{*}Corresponding author

Email address: xjzheng@xidian.edu.cn (Xiaojing Zheng)
experimental and computational data in the literature.

*Keywords:*
Hybrid parallel approach, fully resolved simulation, particle-laden flows, sediment transport, memory optimization
1. Introduction

When an erodible sediment bed is exposed to a sufficiently strong shear flow, bed particles may be entrained and transported by the flow under the combined action of the hydrodynamic force, gravitational force, and interparticle contact force, a process known as sediment transport (Bagnold, 1941; Graf, 1984; Chien and Wan, 1999; Shao, 2008; Zheng, 2009). Sediment transport by wind or water is ubiquitous in nature, such as in sand/dust storms, debris flows, rivers, and coastal environments. It is one of the most important geophysical processes responsible for wind erosion, dust aerosol emission, and the formation of dunes and ripples. A deep understanding of the mechanism of particle-fluid interactions over erodible sediment beds is vital for accurate prediction of sediment transport and geomorphological variations.

There have been extensive experimental and numerical simulation studies on sediment transport over erodible sediment beds (Merritt et al., 2003; Le Roux and Rojas, 2007; Papanicolaou et al., 2008; Durán et al., 2011; Kok et al., 2012; Valance et al., 2015; Yang et al., 2017; Zhang et al., 2018; Zhu et al., 2019; Pähtz et al., 2020; Rana et al., 2021). The interactions between the particle-wall (P-W) process and turbulence in sediment transport have often been the focus of studies (Ho et al., 2011; Lanigan et al., 2016; Bo et al., 2017; Berk and Coletti, 2020; Bragg et al., 2021; Zheng et al., 2021a,b; Zhu et al., 2021). The complex physics of these interactions in sediment transport calls for high-fidelity simulation data. Among different numerical simulation methods, including the Eulerian method, Lagrangian point-particle method, and fully resolved method, the fully resolved direct numerical simulation has the highest fidelity (Uhlmann, 2005; Luo et al., 2007; Yu and Shao, 2007; Breugem, 2012; Kempe and Fröhlich, 2012b; Tenneti and Subramaniam, 2014; Zhou and Fan, 2014; Picano et al., 2015; Akiki and Balachandar, 2016; Tschisgale et al., 2017; Wang et al., 2017a; Costa et al., 2018; Tao et al., 2018; Peng et al., 2019; Wang et al., 2019). This method is also the most computationally intensive method because it needs to fully resolve the scales of turbulence and particles simultaneously. Moreover, in sediment transport, particle collisions need to be modeled, which significantly increases the computational complexity. Benefiting from the advancements in high-performance computation in recent years, fully resolved direct numerical simulations of sediment transport have become feasible (Ji et al., 2014; Vowinckel et al., 2014; Kidanemariam and Uhlmann, 2014a,b, 2017; Jain et al., 2021). However, the high computational cost and memory consumption of this approach pose
great challenges to its utilization in basic and applied research. Further improvements of the parallel algorithms and memory optimization techniques are critically needed.

Sediment transport over an erodible sediment bed can be modeled by coupling computational fluid dynamics (CFD) and the discrete element method (DEM). While there have been extensive studies on the parallelization of individual modules, coupled parallel algorithms for the CFD-DEM model are still limited. Most CFD-DEM parallel algorithms employ the domain decomposition method for the carrier phase. For the disperse phase, the algorithms can be divided into two main categories: (1) parallelization based on particle location, including the domain decomposition method (Uhlmann, 2004; Tsuji et al., 2008; Gopalakrishnan and Tafti, 2013; Wang et al., 2017b; Pozzetti et al., 2019; Dufresne et al., 2020), and (2) parallelization based on particle number, including the particle subset method (Kafui et al., 2011) and the mirror domain technique (Darmana et al., 2006).

In the domain decomposition method, each processor handles a specific subdomain and synchronizes the data across the subdomain boundaries. Tsuji et al. (2008) used a one-dimensional domain decomposition method on a structured grid. They reported a parallel efficiency of 57.8% for the simulation of a fluidized bed using 64 processors. Gopalakrishnan and Tafti (2013) employed the domain decomposition method in the MFiX-DEM code and developed a parallel algorithm for arbitrary 3D unstructured meshes with a parallel efficiency of 32% for the simulation of a rectangular bubbling fluidized bed on 256 processors. Wang et al. (2017b) proposed a parallelization strategy for a lab-scale full-loop circulating fluidized bed based on the domain decomposition method. They obtained a parallel efficiency of 57% on 64 processors. Recently, Pozzetti et al. (2019) developed a dual-grid multiscale parallel strategy based on the domain decomposition method. They obtained good scalability for the parallel efficiency when simulating the motions of ten million point particles in the presence of carrier gas. The parallel efficiency was 73% using 560 processors. Dufresne et al. (2020) incorporated a two-level domain decomposition method into the code YALES2 (Moureau et al., 2011) with unstructured meshes. They reported a good parallel performance from 64 to 1024 processors when simulating a fluidized bed with ten million point particles. The parallel efficiency was 80% using 1024 processors.

In the particle subset method, the particles are evenly distributed to each processor. But the flow data need to be mirrored to all processors when calculating particle-fluid interactions in this method. Kafui et al. (2011) adopted
the particle subset method and distributed the particles using a graph partitioning algorithm. They obtained a parallel efficiency of 57.8% for the simulation of a rectangular powder bed using 64 processors. In the mirror domain technique, each processor stores the same total particle data but only uses a specific subset of them with almost the same particle number. Synchronization of the total particle data is performed after every time step. When calculating particle-fluid interactions, the flow data are separately stored on different processors based on spatial location, i.e., using domain decomposition, while particles are mirrored to all processors, which is contrary to the particle subset method. Darmana et al. (2006) employed domain decomposition for the carrier phase and the mirror domain technique for the disperse phase. They reported a parallel efficiency of 62.5% for the simulation of a buoyancy-driven flow in a square bubble column with 32 processors.

The CFD-DEM parallelization studies reviewed above all used the point-particle approximation. The main difference between the point particle and fully resolved particle methods lies in the treatment of particle-fluid interactions. In the point-particle method, the particle is modeled as a mathematical sizeless point. In the fully resolved simulation, the surface of a finite-sized particle is represented by Lagrangian points, and particle-fluid interactions are usually calculated by the immersed boundary (IB) method or the fictitious domain method (Uhlmann, 2005; Yu and Shao, 2007). The feedback force of a particle on the fluid is determined by interpolating the forces on the Lagrangian points to the Eulerian grid cells. The force can also affect other Eulerian grid cells surrounding the one where the Lagrangian point is located. If the domain decomposition method is employed, then the force may act on a ghost cell of another subdomain when a Lagrangian point is near the subdomain boundary. In this case, the force in the ghost cell needs to be mirrored back to the adjacent subdomain, which increases the complexity of the parallelization of the fully resolved simulation compared with the point-particle simulation. On the other hand, for parallel algorithms of the sediment transport problem, particles are usually distributed unevenly in space and concentrated near the bottom due to gravity. The distribution of particles also varies with time owing to particle movement. Therefore, it is challenging to balance the workload among different processors using the domain decomposition method for the disperse phase. To the best of our knowledge, only Uhlmann (2004) reported the parallel performance of a fully resolved simulation code. They employed a variant of the domain decomposition method, the master-slave technique, to improve the workload balance.
among different subdomains. The parallel efficiency was only 6% for a coarse grid and 40% for a fine grid for the simulation of 48 particles settling in an ambient container using 16 processors. They explained that the low parallel efficiency was caused by the uneven distribution of the particles among processors.

The main advantage of the particle-based decomposition method, i.e., the mirror domain technique and particle subset method, is the good load balance performance for an arbitrary distribution of particles. In a fully resolved simulation, because a particle is discretized such that it contains hundreds of Eulerian grid cells, the communication cost of the mirror flow data is much higher than that of the mirror particles. Therefore, the mirror domain technique may achieve excellent load balance for arbitrary distributions of particles and have a comparatively low communication cost, i.e., as a more efficient method in the parallelization of the disperse phase for sediment transport problems with fully resolved simulations.

In the present study, a hybrid parallelization approach, namely, the domain decomposition method for the carrier phase and a modified mirror domain technique for the disperse phase, is proposed for fully resolved direct numerical simulation (FRDNS) of flows laden with finite-sized particles (for the first time). Furthermore, considering that a large memory may be required in FRDNS, we develop a memory optimization technique for spherical particles of equal size, which can significantly reduce the memory requirement in parallelization. The accuracy of our FRDNS code and the parallel efficiency of the hybrid parallelization approach are rigorously tested. The results indicate that the FRDNS code has high parallel performance and is highly accurate in simulating turbulent flow over an erodible sediment bed. It can serve as a useful tool for studying the interactions between the P-W process and turbulence in sediment transport.

The paper is arranged as follows. The mathematical formulations, models and numerical schemes employed in this work are introduced in §2. The proposed hybrid parallel approach and memory optimization are described in detail in §3. We demonstrate the accuracy and efficiency of the code in §4 through the case of turbulent flow over an erodible sediment bed. The final conclusion of the paper is drawn in §5.
2. Mathematical formulations, models, and numerical schemes

In this section, we introduce the governing equations of the particle-laden turbulent flow, the fluid-particle coupling model, the hydrodynamic force and torque model and the collision model. These models have been well documented in the literature, and we introduce them here just for the completeness of the paper. In addition, several benchmark cases are tested to demonstrate the accuracy of the code.

2.1. Governing equations

The particle-laden flow considered here is governed by the Navier-Stokes equations for the carrier phase and Newton-Euler equations for the disperse particulate phase. The motion of an incompressible, Newtonian fluid flow is governed by the following continuity and momentum equations:

\[ \nabla \cdot \mathbf{u} = 0, \]  
\[ \frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{u} \mathbf{u}) - \frac{1}{\rho_f} \nabla p + \nu_f \nabla^2 \mathbf{u} + \mathbf{f}, \]  

where \( \mathbf{u} \) is the fluid velocity, \( p \) is the dynamic pressure, \( \mathbf{f} \) is the volume force, \( \rho_f \) is the fluid density, and \( \nu_f \) is the fluid kinematic viscosity.

For the numerical discretization of equations (1-2), a second-order central difference scheme is used for spatial discretization, and a second-order Runge-Kutta (RK2) method is used for fluid time advancement (Yang et al., 2017, 2018; Cui et al., 2018). At each substep of the RK2 method, the fractional-step method of Kim and Moin (1985) is applied to ensure that the flow velocity is divergence free. The discretized equations for each Runge-Kutta substep are written as follows:

\[
do k = 1, 2 \]
\[ \hat{\mathbf{u}}^k = \mathbf{u}^{k-1} + \Delta t_f \left[ \alpha_k \mathbf{H}_{k-1} - \beta_k \left( \mathbf{H}^{k-2} - \frac{1}{\rho_f} \nabla p^{k-2} \right) \right] \]
\[ \tilde{\mathbf{u}}^k = \hat{\mathbf{u}}^k + \alpha_k \Delta t_f \mathbf{f}^k \]
\[ \nabla^2 p^{k-1} = \frac{\rho_f}{\alpha_k \Delta t_f} \nabla \cdot \hat{\mathbf{u}}^k \]
\[ \mathbf{u}^k = \tilde{\mathbf{u}}^k - \frac{\alpha_k \Delta t_f}{\rho_f} \nabla p^{k-1} \]
In the above equations, the superscript \( k \) is the index of the Runge-Kutta substep. The coefficients in the RK2 scheme are \( \alpha_1 = 1, \beta_1 = 0, \) and \( \alpha_2 = \beta_2 = 0.5. \) \( \Delta t_f \) is the time step for the fluid solver, \( \hat{u}^k \) and \( \tilde{u}^k \) are the intermediate fluid velocities, \( H^k = -\nabla \cdot (u^k u^k) + \nu_f \nabla^2 u^k \) is the summation of the convection and viscous terms, and \( f^k \) is the volume force from particles computed by the immersed boundary method, which will be introduced in section 2.2. The pressure field is given by the solution of the Poisson equation (5), which is solved using the PETSc library (Balay et al., 1997).

The translational and angular velocities of a particle are solved by the Newton-Euler equations. For a spherical particle, the equations reduce to

\[
\rho_p V_p \frac{du_p}{dt} = \rho_f \int_{\partial V} \tau \cdot n_p dA + (\rho_p - \rho_f) V_p g + F_{c,p}, \tag{7}
\]

\[
I_p \frac{d\omega_p}{dt} = \rho_f \int_{\partial V} r \times (\tau \cdot n_p) dA + T_{c,p}, \tag{8}
\]

where \( \rho_p \) is the particle density, \( V_p \) is the volume of the particle and equals \((4/3)\pi R_p^3\) for a spherical particle with radius \( R_p, \) \( I_p \) is the moment of inertia of the particle and equals \((2/5)\rho_p V_p R_p^2\) for a spherical particle, \( u_p \) and \( \omega_p \) are the translational and angular velocities of the particle, respectively, \( \tau = -pI + \mu_f (\nabla u + \nabla u^T) \) is the hydrodynamic stress tensor (the superscript \( T \) indicates transposition of a tensor), \( n_p \) is the outward-pointing unit normal vector at the particle surface denoted by \( \partial V, \) \( r \) is the position vector of the particle surface relative to the particle center, \( g \) is the gravitational acceleration, and \( F_{c,p} \) and \( T_{c,p} \) are the collision force and torque acting on the particle, respectively. The subscript \( p \) indicates the quantities of particle \( p. \) The discretization of the Newton-Euler equations is introduced in section 2.3.

### 2.2. Fluid-particle coupling model

The fluid and particles are coupled by the direct-forcing immersed boundary method (Uhlmann, 2005), which is a variant of the more general immersed boundary method (Mittal and Iaccarino, 2005; Huang and Tian, 2019). The Lagrangian points are distributed on the surface of a particle, and the fluid domain is discretized by a fixed Cartesian grid with uniform grid spacings in all three directions, i.e., \( \Delta x = \Delta y = \Delta z. \) The interpolated flow velocity at the Lagrangian points on the particle surface may not satisfy the no-slip
boundary condition. Therefore, a volume forcing is enforced at the Eulerian grid points in the vicinity of the Lagrangian points. The volume force $f$ on the right-hand side of the momentum equation (2) is calculated by a multi-direct-forcing immersed boundary method (Luo et al., 2007; Breugem, 2012). This scheme computes the volume force on the Eulerian grid points iteratively to reduce the no-slip boundary condition error at the particle surface. The multi-direct-forcing immersed boundary method works as follows:

$$\begin{align*}
do s = 1, N_s \\
U_l^{s-1} &= \sum_{ijk} u_{ijk}^{s-1} \delta_d (x_{ijk} - X_l^n) \Delta x \Delta y \Delta z \\
F_{p,l}^{s-1} &= \frac{U_p(X_l^n) - U_l^{s-1}}{\Delta t} \\
f_{ijk}^{s-1} &= \sum_l F_{p,l}^{s-1} \delta_d (x_{ijk} - X_l^n) \Delta V_i \\
u^s &= u^{s-1} + \Delta t f^s \\
F_{p,l}^{n,s} &= F_{p,l}^{n,s-1} + F_{p,l}^{s-1}
\end{align*}$$

where the capital letters represent the quantities defined on the Lagrangian points (i.e., the points on the particle), while the lowercase letters represent the quantities defined on the Eulerian grid. The subscript $ijk$ indicates the quantities on the Eulerian grid point with index $(i, j, k)$, the subscript $l$ indicates the quantities on the Lagrangian point with index $l$, and the superscripts $n$ and $s$ indicate the flow solver time step number and the IB iteration number, respectively. $N_s$ is the total IB iteration number in a fluid time step, $x$ and $X$ denote point coordinates, $u$ and $U$ denote flow velocities, $f$ and $F$ denote IB volume forces, $\Delta V_i$ is the volume of a Lagrangian point, and $\Delta t$ is the flow time step. $\delta_d$ is the regularized Dirac delta function, for which we implement the three-point regularized Dirac delta function of Roma et al. (1999), and we choose $N_s = 3$ according to Breugem (2012).

The velocity $U_p$ on a Lagrangian point can be determined by the translational velocity $u_p$ and angular velocity $\omega_p$ of the particle as

$$U_p(X_l^k) = u_p^n + \omega_p^k \times r_l^k,$$

where $r_l^k = X_l^k - x_p^k$ is the relative position vector of the Lagrangian point with respect to the particle center.
2.3. Hydrodynamic force and torque model

It is usually challenging to directly calculate the hydrodynamic force and torque using the surface integrals of equations (7) and (8). To address this issue, Uhlmann (2005) proposed a method by means of momentum balance over the volume occupied by a particle. The momentum balance of the fluid occupied by particle volume $V_p$ is given by

$$\rho_f \oint_{\partial V} \tau \cdot \mathbf{n}_p dA = -\rho_f \int_{V_p} \mathbf{f} dV + \rho_f \frac{d}{dt} \left( \int_{V_p} \mathbf{u} dV \right), \quad (15)$$

$$\rho_f \oint_{\partial V} \mathbf{r} \times (\tau \cdot \mathbf{n}_p) dA = -\rho_f \int_{V_p} \mathbf{r} \times \mathbf{f} dV + \rho_f \frac{d}{dt} \left( \int_{V_p} \mathbf{r} \times \mathbf{u} dV \right). \quad (16)$$

By replacing the surface integral in particle equations of motion (7) and (8), we obtain

$$\rho_p V_p \frac{d\mathbf{u}_p}{dt} = \rho_f \frac{d}{dt} \left( \int_{V_p} \mathbf{u} dV \right) - \rho_f \int_{V_p} \mathbf{f} dV + (\rho_p - \rho_f) V_p \mathbf{g} + \mathbf{F}_{c,p}, \quad (17)$$

$$I_p \frac{d\omega_p}{dt} = \rho_f \frac{d}{dt} \left( \int_{V_p} \mathbf{r} \times \mathbf{u} dV \right) - \rho_f \int_{V_p} \mathbf{r} \times \mathbf{f} dV + \mathbf{T}_{c,p}. \quad (18)$$

For the volume integrals of the first term on the right-hand side of equations (17) and (18), we adopt the numerical evaluation methodology of Kempe and Fröhlich (2012b). They evaluated the volume integrals directly by means of a second-order accurate midpoint rule. Therefore, the two volume integrals are computed as

$$\int_{V_p} \mathbf{u} dV = \sum_{ijk} \mathbf{u}_{ijk} \alpha_{ijk} \Delta x \Delta y \Delta z, \quad (19)$$

$$\int_{V_p} \mathbf{r} \times \mathbf{u} dV = \sum_{ijk} \mathbf{r}_{ijk} \times \mathbf{u}_{ijk} \alpha_{ijk} \Delta x \Delta y \Delta z. \quad (20)$$

Here, $\mathbf{r}_{ijk} = \mathbf{x}_{ijk} - \mathbf{x}_p$ is the relative position vector of the Eulerian grid with respect to the particle center, and $\alpha_{ijk}$ is the volume fraction of a particle in
the Eulerian grid cell with index \((i, j, k)\), which can be calculated approximately by the signed-distance level-set function \(\phi\) of the particle surface \(\partial V\) with \(\phi < 0\) inside and \(\phi > 0\) outside the particle as

\[
\alpha_{ijk} = \frac{\sum_{n=1}^{8} (-\phi_n H(-\phi_n))}{\sum_{n=1}^{8} |\phi_n|},
\]

where the summation is over all eight corner points of the Eulerian grid volume and \(H\) is the Heaviside step function.

Because the regularized Dirac delta function conserves linear and angular momentum, the integrals over the Eulerian forces in equations (17) and (18) can be replaced by the corresponding summations over the Lagrangian forces. Thus, the second terms on the right-hand side of equations (17) and (18) are computed as

\[
\rho_f \int_{V_p} f dV = \rho_f \sum_{l=1}^{N_l} F_l \Delta V_l,
\]

\[
\rho_f \int_{V_p} r \times f dV = \rho_f \sum_{l=1}^{N_l} (r_l \times F_l) \Delta V_l.
\]

Based on the above operations, the particle equations of motion (3) and (4) can be finally rewritten as

\[
\rho_p V_p \frac{d\mathbf{u}_p}{dt} = \rho_f \frac{d}{dt} \left( \sum_{ijk} \mathbf{u}_{ijk} \alpha_{ijk} \Delta x \Delta y \Delta z \right) - \rho_f \sum_{l=1}^{N_l} F_l \Delta V_l + (\rho_p - \rho_f) V_p \mathbf{g} + \mathbf{F}_{c,p},
\]

\[
I_p \frac{d\mathbf{\omega}_p}{dt} = \rho_f \frac{d}{dt} \left( \sum_{ijk} \mathbf{r}_{ijk} \times \mathbf{u}_{ijk} \alpha_{ijk} \Delta x \Delta y \Delta z \right) - \rho_f \sum_{l=1}^{N_l} (r_l \times F_l) \Delta V_l + \mathbf{T}_{c,p}.
\]

2.4. Collision model

Based on previous studies (Kempe and Fröhlich, 2012a; Costa et al., 2015; Biegert et al., 2017), the collision model, which contains the lubrication force and contact force in a fully resolved simulation, is divided into three stages depending on the distance between the colliding objects, as shown in Fig. 1. The stages are the following. (a) When \(\varepsilon \geq \varepsilon_{\Delta x}\), the hydrodynamic force is
fully resolved by the IB method, and no contact occurs. Here, \( \varepsilon = \delta_n / R_p \) is the nondimensional gap width, \( \delta_n \) is the distance between the two surfaces as defined in Appendix A, \( R_p \) is the particle radius and \( \varepsilon_{\Delta x} \) is a predefined constant. (b) When \( 0 \leq \varepsilon < \varepsilon_{\Delta x} \), the IB method cannot fully resolve the hydrodynamic force due to the lack of spatial grid resolution. As a remedy, the lubrication force is added to compensate for the hydrodynamic force. However, no contact occurs at this time. (c) When \( \varepsilon < 0 \), contact occurs. The contact force governs the particle motion, and the hydrodynamic force is not considered.

\[
\begin{align*}
F_{c,p} = \begin{cases} 
0, & \varepsilon \geq \varepsilon_{\Delta x}, \\
\sum_{p,q \neq p}^{N_p} F^{\text{lub}}_n, & 0 \leq \varepsilon < \varepsilon_{\Delta x}, \\
\sum_{p,q \neq p}^{N_p} (F^{\text{col}}_n + F^{\text{col}}_t), & \varepsilon < 0,
\end{cases}
\end{align*}
\]

(26)

where \( N_p \) is the number of particles, \( F^{\text{lub}}_n \) is the lubrication force acting on particle \( p \), and \( F^{\text{col}}_n \) and \( F^{\text{col}}_t \) are the normal and tangential collision forces acting on particle \( p \), respectively.

Figure 1: Sketch of the collision model.
The torque $T_{c,p}$ acting on particle $p$ generated by the tangential contact force is

$$T_{c,p} = \sum_{p,q \neq p} N_p R_p n \times F_{t}^{col},$$

where $n$ is the normal unit vector of contact as defined in Appendix A. The following subsections demonstrate how $F_{n}^{lub}$, $F_{n}^{col}$, and $F_{t}^{col}$ are modeled.

### 2.4.1. Lubrication force model

As mentioned above, at $0 \leq \varepsilon < \varepsilon_{\Delta x}$, a lubrication force is needed to compensate for the hydrodynamic force owing to the lack of spatial grid resolution. A number of lubrication models based on the asymptotic expansion of the analytical solution for the lubrication force in the Stokes regime have been proposed (Brenner, 1961; Cox and Brenner, 1967; Cooley and O’neill, 1969; Kempe and Fröhlich, 2012a; Costa et al., 2015). Here, we employ the two-parameter lubrication model proposed by Costa et al. (2015). The nondimensional gap width $\varepsilon$ is used to evaluate the lubrication force as

$$F_{n}^{lub} = \begin{cases} -6\pi \mu_f R_p u_{cp,n} (\lambda(\varepsilon) - \lambda(\varepsilon_{\Delta x})), & \varepsilon_{\sigma} \leq \varepsilon < \varepsilon_{\Delta x}, \\ -6\pi \mu_f R_p u_{cp,n} (\lambda(\varepsilon_{\sigma}) - \lambda(\varepsilon_{\Delta x})), & 0 \leq \varepsilon < \varepsilon_{\sigma}, \\ 0, & \text{otherwise}, \end{cases}$$

where $\mu_f$ is the fluid dynamic viscosity, $u_{cp,n}$ is the normal component of the relative velocity as defined in Appendix A, $\lambda$ is the Stokes amplification factor for the lubrication interaction, and the subscripts $pq$ and $pw$ indicate the interactions between two particles and between a particle and a wall, respectively. $\varepsilon_{\sigma}$ is a threshold value related to the typical size of the asperities and is fixed to $\varepsilon_{\sigma} = 0.001$. Following Costa et al. (2015), we adopt $\varepsilon_{\Delta x} = 0.025$ for the particle-particle collision and $\varepsilon_{\Delta x} = 0.075$ for the particle-wall collision in this work.

$$\lambda_{pq}(\varepsilon) = \frac{1}{2\varepsilon} - \frac{9}{20} \ln \varepsilon - \frac{3}{56} \varepsilon \ln \varepsilon + O(1),$$

$$\lambda_{pw}(\varepsilon) = \frac{1}{\varepsilon} - \frac{1}{5} \ln \varepsilon - \frac{1}{21} \varepsilon \ln \varepsilon + O(1),$$

where $\lambda_{pq}$ and $\lambda_{pw}$ are the Stokes amplification factors for particle-particle and particle-wall interactions, respectively.
2.4.2. Contact force model

The adaptive collision time model (ACTM) is used to account for contact forces following previous works (Kempe and Fröhlich, 2012a; Biegert et al., 2017). This model is a variation of the traditional soft-sphere model, which is composed of a nonlinear spring-dashpot system in the normal direction and a linear spring-dashpot system with a Coulomb friction slider in the tangential direction. The basic idea of the ACTM is to stretch the collision time $T_c = N \Delta t_f$ and adaptively calibrate the normal stiffness and damping coefficient $k_n$ and $d_n$ based on the impact velocity $u_{in}$. The normal contact force $F_n^{col}$ is determined by

$$F_n^{col} = -k_n |\delta_n|^{3/2} n - d_n u_{cp,n},$$

(31)

where $\delta_n$ and $u_{cp,n}$ are the normal displacement of the spring and normal component of the relative velocity, respectively, as defined in Appendix A. The nonlinear term $|\delta_n|^{3/2}$ arises from Hertzian contact theory (Hertz, 1882). Kempe and Fröhlich (2012a) determined $k_n$ and $d_n$ by an iterative procedure, whereafter Biegert et al. (2017) simplified the calculation by an explicit formulation proposed by Ray et al. (2015). Here, we implement the explicit formulation of Ray et al. (2015), in which $k_n$ and $d_n$ are

$$k_n = \frac{m_{e,n}}{\sqrt{u_{in} t^*}^5}, \quad d_n = \frac{2 \lambda m_{e,n}}{t^*},$$

(32)

where $m_p$ and $m_q$ are the masses of particles $p$ and $q$, respectively, and $m_{e,n} = (m_p^{-1} + m_q^{-1})^{-1}$ is the effective mass for the normal direction. The impact velocity $u_{in} = u_{cp,n} \cdot n$ is recorded at the first occurrence of $\delta_n < 0$, $\lambda = (-C \eta/2 + \sqrt{\frac{1}{4}C^2 \eta^2 + \alpha^2 \tau^2_{c,0}})/\alpha \tau^2_{c,0}$, $t^* = T_c \sqrt{1 - A \lambda - B \lambda^2 / \tau^2_{c,0}}$, where $A = 0.716$, $B = 0.830$, $C = 0.744$, $\alpha = 0.111$, and $\tau_{c,0} = 3.218$. The parameter $\eta = (\ln e_{n,d})^2$ accounts for the dry restitution coefficient in the normal direction, and the collision time is set to $T_c = 10 \Delta t_f \ (N = 10)$ following the suggestion of Kempe and Fröhlich (2012a).

The tangential contact force $F_t^{col}$ is modeled by

$$F_t^{col} = -k_t \delta_t - d_t u_{cp,t},$$

(33)

where $k_t$ and $d_t$ are the tangential stiffness and damping coefficient, respectively, and $\delta_t$ and $u_{cp,t}$ are the tangential displacement of the spring and
normal component of the relative velocity, respectively, as defined in Appendix A. According to Costa et al. (2015), $k_t$ and $d_t$ can be computed by

\[
k_t = \frac{m_{e,t} \left( \pi^2 + \ln^2 e_t \right)}{T_c}, \quad d_t = -\frac{2m_{e,t} \ln e_t}{T_c},
\]

(34)

where $m_{e,t} = (1 + 1/K^2)^{-1}m_{e,n}$ is the effective mass for the tangential direction, and $K$ is the normalized particle radius of gyration, with $K^2 = 2/5$ for a homogeneous sphere particle.

When $|F_{col}^t| \geq \mu_c |F_{col}^n|$ ($\mu_c$ is the friction coefficient), a particle starts sliding. The tangential force is controlled by the slider in the tangential direction based on Coulomb’s friction law instead of the tangential displacement as

\[
F_{col}^t = \mu_c |F_{col}^n| t.
\]

(35)

The tangential displacement needs to be reset to comply with Coulomb’s friction law, that is,

\[
\delta_t = -\frac{\mu_c |F_{col}^n| t + d_t u_{cp,t}}{k_t}.
\]

(36)

2.4.3. Enduring contact model

Although excluding the hydrodynamic force during the collision may yield good agreement with the experiment of Gondret et al. (2002), as shown in Fig. 4, it may also lead to nonphysical scenarios for enduring contact. For example, in a turbulent flow over an erodible sediment bed, the particles on the surface of the bed can be entrained into the flow. If we exclude the hydrodynamic force during the collision, then the surface particles will be stabilized on the bed because they are in contact with other particles in the bed, and no particle can be entrained. Kempe et al. (2014) addressed this problem by turning on the hydrodynamic force for all collisions regardless of $St$. Biegert et al. (2017) found that switching on the hydrodynamic force for collision with high $St$ would underpredict the particle rebounding trajectory. Furthermore, they investigated the critical $St$ for excluding the hydrodynamic force and demonstrated that the hydrodynamic force needs to be considered when $St \leq 5$.

We adopt the enduring contact model of Biegert et al. (2017), in which the particle translational velocity equation (7) can be divided into three
situations as
\[
\rho_p V_p \frac{d u_p}{d t} =
\begin{cases}
\rho_f \oint_{\partial V} \tau \cdot n_p dA + (\rho_p - \rho_f) V_p g + F_{c,p}, & \delta_n > 0.
\end{cases}
\]

The particle angular velocity equation (8) is treated in the same way.

2.5. Validation

In this part, we validate the accuracy of the immersed boundary method and collision model step by step. Three cases are simulated and compared with numerical or experimental results in the literature.

2.5.1. A fixed spherical particle in uniform cross flows

To validate the accuracy of the IB method, a fixed spherical particle in uniform cross flows is simulated. A spherical particle is placed at the center of the domain. The dimensions of the domain are $15D_p \times 6D_p \times 6D_p$ resolved by a $300 \times 120 \times 120$ grid (20 grid points in $D_p$) along the streamwise, vertical, and spanwise directions, where $D_p$ is the diameter of the particle. Inflow and outflow boundary conditions are implemented at the inlet and outlet, respectively, and periodic boundary conditions are enforced at the side boundaries. The computed drag coefficients at different particle Reynolds numbers (defined as $Re_p = U_\infty D_p / \nu_f$, $U_\infty$ is the inflow velocity) are shown in Fig. 2. The solid line is the empirical drag law (the S-N law), $C_D = (24/Re_p)(1 + 0.15Re_p^{0.687})$, which was proposed by Schiller and Naumann (1933). It is seen from Fig. 2 that the computed results are in good agreement with the S-N law, which validates the present implementation of the multi-direct-forcing immersed boundary method.
2.5.2. Sedimentation of a spherical particle in a quiescent fluid

Next, we validate the accuracy of the IB method for a moving particle using the case of the sedimentation of a spherical particle in a quiescent fluid with different density ratios. The computation configuration is the same as that in Uhlmann (2005), and the parameters used here are listed in Table 1. Sufficiently large domain size and periodic boundary conditions are applied in all three directions to match the experiment of Mordant and Pinton (2000). The computational domain size is $L_x \times L_y \times L_z = 7.68D_p \times 54D_p \times 7.68D_p$ with a grid of $N_x \times N_y \times N_z = 128 \times 1024 \times 128$ (16.7 grid points in $D_p$). The particle is initially placed at $x = L_x/2$, $y = 0.9L_y$, $z = L_z/2$ and released under gravity from rest at $t = 0$. The computed settling velocity is displayed in Fig. 3. Compared with the measurement data of Mordant and Pinton (2000) and the simulation results of Uhlmann (2005), it is seen that the present results under different density ratios are all in good agreement with them, validating the current code for moving particles.
Table 1: Parameters used in the simulation of spherical particle sedimentation.

| \(Re_p\) | \(\rho_p/\rho_f\) | \(g\) | \(D_p\) | \(nu\) | \(D_p/\Delta x\) | \(\Delta t\) |
|---------|-----------------|------|--------|------|----------------|--------|
| 377     | 2.56            | 9.81 | 0.167  | 1.04 \(\times\) 10\(^{-3}\) | 16.7   | \(10^{-3}\)   |
| 283     | 7.71            | 9.81 | 0.167  | 2.68 \(\times\) 10\(^{-3}\) | 16.7   | \(10^{-3}\)   |

Figure 3: Comparison of the settling velocity for different density ratios: (a) \(\rho_p/\rho_f = 2.56\) and (b) \(\rho_p/\rho_f = 7.71\). The settling velocity and the time are normalized by \(u_{ref} = \sqrt{D_p |g|}\) and \(t_{ref} = \sqrt{D_p/(g)}\), respectively.

2.5.3. Normal particle-wall collision in a viscous fluid

To validate the accuracy of the collision model, the bouncing motion of a single particle in a viscous fluid with different Stokes numbers is simulated. The computational configuration is the same as that in Biegert et al. (2017), and the parameters used in the simulation are listed in Table 2.

A periodic boundary condition is imposed in the streamwise and spanwise directions, and a no-slip boundary condition is imposed on both the top and bottom surfaces. The particle is initially placed at \(x = L_x/2\), \(y = L_y - 0.75D_p\), \(z = L_z/2\). We prescribe the falling velocity of the particle following Biegert et al. (2017), that is, accelerate it smoothly and let \(u_{in}\) match the Stokes number in the experiment (Gondret et al., 2002) before the collision, as

\[
u_p(t) = u_{in}\left(e^{-40t} - 1\right), \quad \text{if} \quad \delta_n > R_p.
\]

Once the particle reaches a distance of \(\delta_n = R_p\), we turn off the prescribed velocity. Then, the particle moves under hydrodynamic, gravitational, buoy-
Table 2: Parameters used in the simulation of the particle-wall collision case.

| Parameter | Value 1 | Value 2 |
|-----------|---------|---------|
| $St$      | 27      | 152     |
| $Re_p$    | 30      | 164     |
| $D_p$ (m) | 0.006   | 0.003   |
| $u_{in}$ (m/s) | 0.519 | 0.585 |
| $\rho_p/\rho_f$ | 8.083 | 8.342 |
| $\nu_f$ (m²/s) | $1.036 \times 10^{-4}$ | $1.070 \times 10^{-5}$ |
| $e_n$     | 0.97    | 0.97    |
| $g$ (m/s²) | 9.81    | 9.81    |
| $L_x \times L_y \times L_z$ (m) | $12.8D_p \times 25.6D_p \times 12.8D_p$ | $6D_p \times 70D_p \times 6D_p$ |
| Grid number | $256 \times 512 \times 256$ | $120 \times 1400 \times 120$ |
| $D_p/\Delta x$ | 20      | 20      |

ant, and collision forces. Figure 4 shows the computed particle trajectories at different Stokes numbers. The results are in good agreement with the experiment of Gondret et al. (2002).

Figure 4: Comparison of the particle trajectories between the experiment and the present simulation for different Stokes numbers: (a) $St = 27$ and (b) $St = 152$.

To summarize section 2.5, we conducted simulations of three test cases to validate the IB method for stationary and moving particle problems and the particle collision model. The results are in good agreement with the experimental and simulation data reported in the literature, confirming the
accuracy of the present particle-resolved direct numerical simulation code. In
the next section, we will introduce the hybrid parallel approach for improving
the computational performance.

3. Hybrid parallel approach and memory optimization

In this section, we present the hybrid parallel approach developed for the
fully resolved simulation of flows laden with finite-sized particles. The carrier
phase is paralleled with the traditional domain decomposition method, which
is commonly used in CFD. Hence, its details are omitted here. The parallel
approaches for the disperse phase and particle-fluid interactions, together
with memory optimization, are the focus of discussion.

3.1. Parallel approach for the disperse phase

As mentioned above, the domain decomposition technique has difficulty
obtaining high parallel efficiency because the spatial distribution of the par-
ticles is often highly uneven, leading to the problem of load imbalance. To
balance the workload among different processors, we modify the mirror do-
main technique proposed by Darmana et al. (2006) for parallelization of the
disperse phase in FRDNS.

Different from the domain decomposition method where each processor
corresponds to a specific subdomain and transmits data at the subdomain
boundaries, each processor in the mirror domain method stores the same
total particle data but only deals with a subset of them. Figure 5 illustrates
the mirror domain technique, where P1 and P2 denote two processors. Each
processor stores the data of all particles. The initial states of all particles in
the two processors are the same, as shown in Fig. 5a. At the next time step,
when the particles move along the arrows to the new positions, as shown
in Fig. 5b, processor P1 only deals with the particles in black, while pro-
cessor P2 deals with the particles in red. After the computation of particle
movements, synchronization is performed to update the data of all parti-
tles in each processor. Then, the same total particle data are obtained and
prepared for the next step of computation, as shown in Fig. 5c. The main
advantage of the mirror domain technique compared with the domain de-
composition technique is that the number of particles stored and computed
in each processor is the same, regardless of the particle distribution, which
can help achieve excellent load balance and improve the parallel efficiency for
the disperse phase. However, it should be noted that the number of particles is limited by the storage size of each processor.

![Image of particle movement and synchronization](image)

**Figure 5:** Illustration of the mirror domain technique for the disperse phase. (a) Initial state of the particles in each processor; (b) particle movement in each processor; and (c) final state of the particles after synchronization in each processor. The black and red particles belong to different particle subsets in P1 and P2, respectively. The blue arrow is the particle movement path.

### 3.2. Parallel algorithm for particle-fluid interactions

Different from the mirror domain technique developed for point-particle simulations (Darmana et al., 2006), the present work develops a modified mirror domain technique for fully resolved simulations. There are three main differences between these two methods:

1. In a fully resolved simulation, a particle is treated as having a finite size by using Lagrangian points on the particle surface (the red points in Figure 6) rather than as a sizeless point in a point-particle simulation. Because the carrier phase and disperse phase are parallelized with different methods, the particle data need to be mirrored to each fluid subdomain. For a point-particle simulation, the subdomain in which a particle is mirrored is unique, while this may not be the case for a fully resolved simulation in which the Lagrangian points of a finite-sized particle may be mirrored to multiple subdomains. For example, for the particle on the bottom left of Fig. 6, even
though the particle center is not in the subdomain, as shown, the particle
data still need to be mirrored to this subdomain, as there are two Lagrangian
points in it.

(2) Because a particle is represented by a finite size rather than as a
sizeless point in the fully resolved simulation, multiple parts of the particle
may be located in different subdomains if the particle is near the subdomain
boundary, as shown in Fig. 6, which does not occur in the point-particle
simulation. The fluid force on the particle is calculated by summing the
forces on all Lagrangian points, and thus, Lagrangian force data transmission
from different fluid subdomains may be needed.

(3) The hydrodynamic force of a particle acting on the fluid is calculated
by diffusing the Lagrangian forces to the surrounding Eulerian grid cells (the
green points in Fig. 6), as described in equation (11). The Lagrangian forces
affect not only the Eulerian grid cell where the Lagrangian point is located
but also the surrounding Eulerian grid cells due to the diffusing function.
For example, as shown in Fig. 6, a diffused Lagrangian force may exist in
the ghost-cell region (the yellow region in Fig. 6) if the Lagrangian point
is near a subdomain boundary. Thus, it is required that the diffused force
in the ghost-cell region be mirrored back to the adjacent subdomain and
superimposed with the existing force.

The above three major differences may significantly increase the com-
plexity of the parallelization of a fully resolved simulation code. The present
work takes into account these issues and extends the traditional mirror do-
main technique from point-particle simulations to fully resolved simulations.
The technique is straightforward; thus, the details are not described here
because of space limitations.
Mirror back

Processor 1

Mirror back

Processor 2

Figure 6: Two-dimensional illustration of data transmission in particle-fluid interaction calculations. The blue circle is the finite-sized particle. The yellow region is the ghost-cell region. The red points are the Lagrangian points on the particle surface. The green points are the Eulerian grid points affected by a Lagrangian point.

3.3. Memory optimization

The available computer memory on high-performance computing platforms limits the number of particles that can be simulated, especially for the mirror domain technique, in which the data of particle quantities, such as the particle position \( \mathbf{x}_p \) and translational and angular velocities \( \mathbf{u}_p \) and \( \omega_p \), need to be stored on each computing node. In addition to the above quantities, five relative quantities are also needed in the collision model, including the relative displacements \( \delta_n \) and \( \delta_t \), the relative velocities \( \mathbf{u}_{cp,n} \) and \( \mathbf{u}_{cp,t} \), and the impact velocity \( u_{in} \), as introduced in section 2.4.2. Although \( \delta_n, \mathbf{u}_{cp,n}, \) and \( \mathbf{u}_{cp,t} \) can be calculated directly from the particle position and velocity without memory consumption, \( \delta_t \) and \( u_{in} \) require large memories as the particle number increases. The memory consumption for \( \delta_t \) and \( u_{in} \) is determined by the number of particle pairs, which equals the square of the particle number. Next, we show that \( \delta_t \) and \( u_{in} \) can be stored with a significantly reduced memory cost.

Here, we propose a novel memory optimization technique to minimize the memory cost of the particle-related variables for spherical particles of the same size. The key idea is to utilize the feature of spherical particles of the same size that each particle can only be in contact with up to twelve surrounding particles in the case of dense packing (Dai et al., 2019). There-
fore, the relative quantities of only the twelve surrounding particles that are in contact need to be stored for each particle considered for collision computation. Without this method, the memory requirement for each relative variable is $N_p^*N_p$, where $N_p^*$ is the number of particles residing in each subset, and $N_p$ is the total particle number. As a result, one can only afford thousands of particles on a common computing platform. By implementing the optimization method developed in this study, the memory cost can be greatly reduced by a factor of $N_p/12$ to $12N_p^*$. As a result, we can handle millions of particles on a common high-performance platform, which is comparable to the highest particle number in the recent fully resolved simulation reported by Kidanemariam and Uhlmann (2017).

Furthermore, the particle collision model requires identifying the collision state between two particles, i.e., whether they are undergoing an existing collision event, a new collision event, or a finished collision event, because different collision events have different operations on $u_{in}$ and $\delta_t$. The surrounding particles in contact with a specific particle usually change with time. For example, as shown in Fig. 7, at time step $n$, particle $p$ is in contact with particle $p-5$, particle $p-1$, and particle $p+3$. However, at time step $n+1$, the contact particles of particle $p$ change to particle $p-1$, particle $p+1$, and particle $p+3$. Therefore, additional quantities and treatments are needed to identify and advance the collision state. Here, we use an array $M_p$ to store the contact particle indexes, as well as the $u_{in}$ and $\delta_t$ of particle $p$ at time step $n$, and a temporary array $M_{temp}$ to store the same kind of data at the last time step $n-1$. Given the above new quantities, three collision events can be identified as follows. (a) If it is a new collision event, that is, a particle collides with particle $p$ at this time step but not at the last time step, then its index is not in $M_{temp}$ (e.g., the red particle indexes $p-5$ and $p+1$ at time steps $n$ and $n+1$, respectively), $u_{in}$ needs to be recorded, and $\delta_t$ needs to be initialized. (b) If it is an existing collision event, i.e., a particle collides with particle $p$ and its index is already in $M_{temp}$ (e.g., the blue particle indexes $p-1$ and $p+3$ at time steps $n$ and $n+1$, respectively), then $u_{in}$ is inherited from $M_{temp}$, and $\delta_t$ is advanced from $M_{temp}$. (c) If a collision event is finished at this time step, i.e., the particle index is in $M_{temp}$ but not $M_p$, then the particle completes the collision (e.g., the green particle indexes $p+5$ and $p-5$ at time steps $n$ and $n+1$, respectively). In this case, $u_{in}$ and $\delta_t$ are reset to zero.

Finally, it is noted that the present memory optimization method can be applied not only in the mirror domain technique but also in other parallel...
methods or point-particle simulations.

![Diagram](image)

**Figure 7:** Sketch of the memory optimization method with the dense packing concept. $M_p$ and $M_{temp}$ are two arrays to store the collision information at time steps $n$ and $n-1$, respectively. The circles in different colors indicate particles in different collision events.

### 3.4. Parallel efficiency

To investigate the parallel efficiency of the proposed hybrid domain decomposition and modified mirror domain methods, we conduct a simulation for a benchmark case where 48 cylindrical particles settle under gravity in a closed ambient container, following Uhlmann (2004). The grid resolution is $N_x \times N_y \times N_z = 512 \times 512 \times 1$, where only one grid cell is kept in the $z$ direction to match the two-dimensional simulation of Uhlmann (2004). The surface of each cylindrical particle is resolved by 30 Lagrangian points. The parallel performance of the approaches can be evaluated by the speed-up ratio $S_n$ and efficiency $E_n$, defined by

$$S_n = \frac{T_1}{T_n}, \quad E_n = \frac{T_1}{N_{CPU}T_n},$$

where $T_1$ and $T_n$ are the execution time of the serial code on a single processor and paralleled code using $N_{CPU}$ processors, respectively.

A comparison of the speed-up ratio and parallel efficiency between Uhlmann (2004) and the present method is shown in Fig. 8. The speed-up ratio and
parallel efficiency of Uhlmann (2004) using the traditional domain decomposition method rapidly decrease as the processor number increases, which can be as low as $S_n \approx 0.9$ and $E_n \approx 6\%$ when using 16 processors due to the uneven distribution of the particles (Uhlmann, 2004). In comparison, the present hybrid parallel method can yield excellent speed-up ratio and parallel efficiency, $S_n \approx 12$ and $E_n \approx 74\%$, when using 16 processors, which are approximately 12 times higher than those in Uhlmann (2004).

Figure 8: Comparison of the speed-up ratio and parallel efficiency for particles settling in a container.

4. Turbulent flow over an erodible sediment bed

4.1. Simulation setup

In this section, we test the parallel efficiency of the code in the computation of a turbulent flow over an erodible sediment bed. The simulation setup is similar to the fully resolved simulation of Ji et al. (2014). The flow is driven by a horizontal body force that is balanced by the shear stress on the sediment bed. The sediment bed consists of $N_p = 4608$ particles with two to three layers, as shown in Fig. 9. In the present study, the sediment bed is generated by a sedimentation DEM simulation for particles settling under gravity while turning off the hydrodynamic force, similar to Kidanemariam and Uhlmann
A periodic boundary condition is imposed in the streamwise and spanwise directions, a no-slip boundary condition is imposed on both the bottom surface and particle surface, and a free-slip boundary condition is imposed on the domain top.

For the carrier phase, the size of the computational domain is $L_x \times L_y \times L_z = (6 \times 1 \times 3)H$ resolved on a uniform Cartesian grid of $N_x \times N_y \times N_z = 960 \times 160 \times 480$, where $H$ is the computational domain height in the $y$ direction. Here, $x$, $y$ and $z$ denote the streamwise, wall-normal and spanwise directions, respectively. The Reynolds number is $Re_{\tau} = u_\tau H/\nu = 678$, where $u_\tau = \sqrt{\tau_b/\rho_f}$ is the friction velocity, $\tau_b$ is the mean shear stress at an effective sediment-bed height $y_b$ (also called the zero-plane displacement), $He = H - y_b$ is the effective boundary-layer height, and the superscript + indicates quantities normalized in viscous units (by $u_\tau$ and $\nu_f$). The effective bed height $y_b$ and equivalent bed roughness $k_s$ are determined by fitting the mean velocity profile of the rough bed flow to the classical logarithmic law (Raupach et al., 1991; Jiménez, 2004; Singh et al., 2007; Ji et al., 2014; Chung et al., 2021; Kadivar et al., 2021), which yields $y_b = 0.228$ and $k_s = 0.224$, which are slightly smaller than the values of $y_b = 0.252$ and $k_s = 0.242$ in Ji et al. (2014), likely due to the different distribution of particles in the
sediment bed. The roughness Reynolds number is \( k_s^+ = 224 \), which indicates a fully rough flow regime. The dimensionless time step for the carrier phase is \( \Delta t_f = 1.5 \times 10^{-4} \).

The parameters for the disperse phase are as follows. The density ratio between the particles and fluid is \( \rho_p/\rho_f = 2.65 \). The dimensionless particle diameter is \( D_p/\nu_f = 0.1 \). The particle diameter in wall units is \( D_p^+ = D_p/(\nu_f/u_\tau) = 88 \). The mean volume fraction of the disperse phase is 13.4%. For the parameters in the collision model, the normal restitution coefficient is \( e_{n,d} = 0.97 \), the tangential restitution coefficient is \( e_{t,d} = 0.39 \) and the friction coefficient is \( \mu_c = 0.15 \), which are chosen according to the material properties of the sand particle (Joseph and Hunt, 2004). The collision time uses \( T_c = 10\Delta t_f \) following the suggestion by Kempe and Fröhlich (2012a), and the dimensionless time step for the particle is \( \Delta t_p = \Delta t_f/50 \). The Shields number is \( \Theta = u_\tau^2/((\rho_p/\rho_f - 1)gD_p) = 0.065 \), and the corresponding Galileo number is \( Ga = \sqrt{(\rho_p/\rho_f - 1)gD_p^2/\nu} = 365 \).

It should be noted that while the simulation model and parameters of the present simulation are matched to those of Ji et al. (2014), there is still a difference in the collision model. We adopt the ACTM (Kempe and Fröhlich, 2012a; Biegert et al., 2017) while Ji et al. (2014) Ji employed the combined finite-discrete element method (Munjiza et al., 1995; Munjiza and Andrews, 2000). In the ACTM, contact forces are modeled by a spring-dashpot system. In the combined finite-discrete element method, contact forces are calculated by the deformation of particles, which is simulated by the finite-element method. This difference may cause different particle distributions in the sediment bed.

### 4.2. Statistics of the carrier and disperse phases

Two cases are considered in the present study. The first case is the turbulent flow over a static sediment bed in which all particles are stationary. The second case is the turbulent flow over a mobile sediment bed in which particles are moved under the actions of hydrodynamic, gravitational and collision forces. To make a comparison with Ji et al. (2014), the wall-normal coordinate \( Y = y - y_b \) is adopted and normalized by \( H_e = H - y_b \) to eliminate the influence of sediment bed height. The effective height of the sediment bed is shown by the gray line in the following figures. The averaging operations for the variables of the carrier and disperse phases are defined in Appendix B.
Figure 10 shows the mean streamwise velocity profile. The operator $\langle \cdot \rangle$ indicates an average over the $x - z$ plane and time. For both the static and mobile cases, our results are generally in good agreement with those of Ji et al. (2014). Above the bed surface, the mean flow velocity in the mobile case is smaller than that in the static case. This is because the entrained particles exert a drag force on the flow and retard the flow accordingly.

![Figure 10: Mean streamwise velocity profile of the carrier phase. The gray line indicates the effective height of the sediment bed.](image)

The root-mean-squared (r.m.s.) velocity profiles of the disperse phase are shown in Fig. 11. The prime superscript indicates the r.m.s. values of fluctuating velocities. The present r.m.s. velocities in the three directions in the static case are in good agreement with those in Ji et al. (2014). Good agreement can also be seen for the mobile case except in the vicinity of the bed surface, where the present r.m.s. values of the carrier phase velocity fluctuations are stronger, especially for the streamwise component. This may be caused by the different particle distributions in the sediment bed or the different collision models noted above.
Figure 11: Profiles of r.m.s. velocities of the carrier phase: (a) static case and (b) mobile case. The gray line indicates the effective height of the sediment bed.

The results of the Reynolds shear stress for the carrier phase are presented in Fig. 12. The present simulation and results in Ji et al. (2014) are in good agreement for both the static and mobile cases.

Figure 12: Reynolds shear stress profiles of the carrier phase. The gray line indicates the effective height of the sediment bed.
For the disperse phase, we compare the results of the nondimensional transport rate $\phi_p$ and the mean velocity profile. The nondimensional transport rate is defined as $\phi_p = \int_0^H q(y) \, dy / \sqrt{(\rho_p/\rho_f - 1) g D_p^3}$, in which $q = C \langle u_p \rangle$ is the volume flux density of the disperse phase, with $C$ being the particle volume fraction and $\langle u_p \rangle$ the mean streamwise particle velocity. The present simulation yields $\phi_p = 0.0331$, slightly higher than the value of $\phi_p = 0.0327$ in Ji et al. (2014). This result is also in good agreement with several empirical bedload transport models and experimental data, with values in the range of 0.01 to 0.04 at $\Theta = 0.065$ (Wiberg and Dungan Smith, 1989). Finally, the mean velocity profiles of the disperse phase are shown in Fig. 13. The mean particle velocities in the three directions are in good agreement with the results in Ji et al. (2014).

![Figure 13: Mean velocity profiles of the disperse phase. The gray line indicates the effective height of the sediment bed.](image)

4.3. Parallel efficiency

The parallel efficiency results of the sediment transport case are presented here. In our simulation, each spherical particle is represented by 805 Lagrangian points. Because of the memory requirement, this simulation needs to be conducted utilizing at least two nodes in the Tianhe-2A supercomputer with 64G memory and 24 processors in one CPU node. Therefore, the
speed-up ratio $S_n$ and parallel efficiency $E_n$ are defined by

$$S_n = \frac{T_{48}}{T_n}, \quad E_n = \frac{T_{48}}{N_{CPU}T_n},$$

in this simulation case, where $T_{48}$ is the execution time of the code using 48 processors. From Fig. 14, it is seen that the present code can achieve strong scalability up to 768 processors, with an efficiency of approximately 90%. The results above indicate that the present hybrid parallel approach has high accuracy and excellent parallel performance for fully resolved simulations of particle-laden turbulent flows in sediment transport.

![Graph showing speed-up ratio and parallel efficiency](image)

Figure 14: Speed-up ratio and parallel efficiency of the simulation of turbulent flow over an erodible sediment bed. The dashed line is the ideal linear scaling for the speed-up ratio.

5. Conclusions

In the present study, a hybrid parallel approach is developed for fully resolved simulations of particle-laden flows and applied to sediment transport of turbulent flow over an erodible sediment bed. The approach combines the domain decomposition method for the carrier phase and a modified mirror domain technique for the disperse phase. By utilizing the strengths of the mirror domain technique in which each processor handles the same number
of particles regardless of the particle distribution, an excellent load balance is achieved compared with the domain decomposition method for the disperse phase. Furthermore, a novel memory optimization method is proposed for spherical particles of the same size, which can substantially reduce the memory cost and enable simulations of millions of fully resolved particles on a common high-performance computing platform.

The present hybrid approach is applied to several benchmark cases, including the sediment transport case of a turbulent flow over an erodible sediment bed. The results show good agreement with the experimental and simulation data in the literature. The present hybrid parallel approach can yield a parallel efficiency of 74%, which is approximately 12 times higher than the results using the domain decomposition method reported for a particle settling case. It also shows excellent scalability, with an efficiency of nearly 90% with up to approximately one thousand processors in the fully resolved simulation of a sediment transport case.

6. Acknowledgments

This work was financially supported by grants from the National Natural Science Foundation of China (Nos. 92052202 and 11972175), the National Numerical Windtunnel project and the Fundamental Research Funds for the Central Universities (lzujbky-2020-17). The computation was partly performed at the Tianhe-2A supercomputer of the National Supercomputer Center in Guangzhou.

Appendix A. Definitions for particle-particle and particle-wall collisions

The variables used in particle collisions are defined below. Some definitions depend on whether the interaction is between particle \( p \) and the wall (P-W) or between particle \( p \) and particle \( q \) (P-P). They are as follows:

- \( n \) - normal unit vector of contact

\[
\begin{align*}
n &= \frac{x_q - x_p}{|x_q - x_p|}, \quad \text{(P-P)}, \\
n &= \frac{x_w - x_p}{|x_w - x_p|}, \quad \text{(P-W)},
\end{align*}
\]
\( \delta_n \) - distance between two surfaces

\[
\delta_n = |x_q - x_p| - R_p - R_q, \quad \text{(P-P),} \quad (A.3)
\]

\[
\delta_n = |x_w - x_p| - R_p, \quad \text{(P-W),} \quad (A.4)
\]

\( \mathbf{u}_{cp} \) - relative velocity of the contact point

\[
\mathbf{u}_{cp} = \mathbf{u}_p - \mathbf{u}_q + R_p \omega_p \times \mathbf{n} + R_q \omega_q \times \mathbf{n}, \quad \text{(P-P),} \quad (A.5)
\]

\[
\mathbf{u}_{cp} = \mathbf{u}_p + R_p \omega_p \times \mathbf{n}, \quad \text{(P-W),} \quad (A.6)
\]

\( \mathbf{u}_{cp,n} \) - normal component of \( \mathbf{u}_{cp} \)

\[
\mathbf{u}_{cp,n} = (\mathbf{u}_{cp} \cdot \mathbf{n}) \mathbf{n}, \quad (A.7)
\]

\( \mathbf{u}_{cp,t} \) - tangential component of \( \mathbf{u}_{cp} \)

\[
\mathbf{u}_{cp,t} = \mathbf{u}_{cp} - \mathbf{u}_{cp,n}, \quad (A.8)
\]

and \( \delta_t \) - tangential displacement of the contact point.

The direction of the tangential unit vector changes at different time steps. Therefore, we need to rotate the displacement from the previous time step onto a plane tangent to \( \mathbf{n} \). \( \delta_t \) is calculated following Biegert et al. (2017) as

\[
\tilde{\delta}_t = \delta_{t}^{k-1} - (\delta_{t}^{k-1} \cdot \mathbf{n}) \mathbf{n}, \quad (A.9)
\]

\[
\hat{\delta}_t = \frac{|\delta_{t}^{k-1}|}{|\delta_t|} \tilde{\delta}_t, \quad (A.10)
\]

\[
\delta_t^k = \hat{\delta}_t + \Delta t \mathbf{u}_{cp,t}. \quad (A.11)
\]

Appendix B. Averaging operations for flow and particle variables

B.1. Averaging operations for flow variables

Before averaging the flow variables, an indicator function needs to be defined as \( \phi_f(x, t) \) to distinguish the Eulerian grid point at a position \( x \) that is inside or outside of a particle, following Kidanemariam and Uhlmann (2014b), as

\[
\phi_f(x, t) = \begin{cases} 
1, & \text{if } x \text{ is outside a particle,} \\
0, & \text{otherwise.} 
\end{cases} \quad (41)
\]
Based on the indicator function $\phi_f(x, t)$, only the flow data outside of the particle are accounted for as follows:

$$n_f(y_j) = \sum_{n=1}^{N_t} \sum_{i=1}^{N_x} \sum_{k=1}^{N_z} \phi_f(x_{ijk}, t^n),$$

where $n_f(y_j)$ is the total grid number in the $x - z$ plane over $N_t$ time steps for flow statistics at a given height $y_j$. Therefore, the ensemble average of the flow variables $\xi_f(x, t)$ can be defined as

$$\langle \xi_f \rangle (y_j) = \frac{1}{n_f(y_j)} \sum_{n=1}^{N_t} \sum_{i=1}^{N_x} \sum_{k=1}^{N_z} \xi_f(x_{ijk}, t^n) \phi_f(x_{ijk}, t^n),$$

where the operator $\langle \cdot \rangle$ indicates the average over the $x - z$ plane and time.

**B.2. Averaging operations for particle variables**

Particle variables are averaged over the particle number within each bin. The bin is generated by dividing $H$ by the thickness $\Delta h$. An indicator function $\phi_p^j(y_p, t)$ is defined to distinguish the center height $y_p$ of a particle inside or outside of a bin with index $j$ as follows:

$$\phi_p^j(y_p, t) = \begin{cases} 1, & \text{if } (j - 1)\Delta h \leq y < j\Delta h \\ 0, & \text{otherwise.} \end{cases}$$

Based on the indicator function $\phi_p^j(y_p, t)$, the particle number in each bin can be calculated as

$$n_p^j = \sum_{n=1}^{N_t} \sum_{l=1}^{N_p} \phi_p^j(y_p^l, t^n),$$

where $n_p^j$ is the total particle number in bin $j$ over $N_t$ time steps. Therefore, the averaged particle variable $\xi_p$ can be defined as

$$\langle \xi_p(y^j) \rangle = \frac{1}{n_p^j} \sum_{n=1}^{N_t} \sum_{l=1}^{N_p} \phi_p^j(y_p^l, t^n) \xi_p^l(t^n).$$

A bin thickness of $\Delta h = D_p/4$ is chosen here. If $n_p^j/N_t < 1$, the $\langle \xi_p(y^j) \rangle$ calculated in bin $j$ is not shown owing to insufficient particle samples.
References

Akiki, G., Balachandar, S., 2016. Immersed boundary method with non-uniform distribution of Lagrangian markers for a non-uniform Eulerian mesh. Journal of Computational Physics 307, 34–59.

Bagnold, R. A., 1941. The physics of blown sand and desert dunes. William Morrow & Company, New York.

Balay, S., Gropp, W. D., McInnes, L. C., Smith, B. F., 1997. Efficient management of parallelism in object-oriented numerical software libraries. In: Modern software tools for scientific computing. Springer, pp. 163–202.

Berk, T., Coletti, F., 2020. Transport of inertial particles in high-Reynolds-number boundary layers. Journal of Fluid Mechanics 903, A18.

Biegert, E., Vowinckel, B., Meiburg, E., 2017. A collision model for grain-resolving simulations of flows over dense, mobile, polydisperse granular sediment beds. Journal of Computational Physics 340, 105–127.

Bo, T., Fu, L., Liu, L., Zheng, X., 2017. An improved numerical model suggests potential differences of wind-blown sand between on Earth and Mars. Journal of Geophysical Research: Atmospheres 122 (11), 5823–5836.

Bragg, A. D., Richter, D. H., Wang, G., 2021. Mechanisms governing the settling velocities and spatial distributions of inertial particles in wall-bounded turbulence. Physical Review Fluids 6 (6), 064302.

Brenner, H., 1961. The slow motion of a sphere through a viscous fluid towards a plane surface. Chemical Engineering Science 16 (3-4), 242–251.

Breugem, W.-P., 2012. A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. Journal of Computational Physics 231 (13), 4469–4498.

Chien, N., Wan, Z., 1999. Mechanics of sediment transport. ASCE Press, Reston.

Chung, D., Hutchins, N., Schultz, M. P., Flack, K. A., 2021. Predicting the drag of rough surfaces. Annual Review of Fluid Mechanics 53, 439–471.
Cooley, M., O’Neill, M., 1969. On the slow motion generated in a viscous fluid by the approach of a sphere to a plane wall or stationary sphere. Mathematika 16 (1), 37–49.

Costa, P., Boersma, B. J., Westerweel, J., Breugem, W.-P., 2015. Collision model for fully resolved simulations of flows laden with finite-size particles. Physical Review E 92 (5), 053012.

Costa, P., Picano, F., Brandt, L., Breugem, W.-P., 2018. Effects of the finite particle size in turbulent wall-bounded flows of dense suspensions. Journal of Fluid Mechanics 843, 450–478.

Cox, R. G., Brenner, H., 1967. The slow motion of a sphere through a viscous fluid towards a plane surface-II small gap widths, including inertial effects. Chemical Engineering Science 22 (12), 1753–1777.

Cui, Z., Yang, Z., Jiang, H.-Z., Huang, W.-X., Shen, L., 2018. A sharp-interface immersed boundary method for simulating incompressible flows with arbitrarily deforming smooth boundaries. International Journal of Computational Methods 15 (1), 1750080.

Dai, W., Reimann, J., Hanaor, D., Ferrero, C., Gan, Y., 2019. Modes of wall induced granular crystallisation in vibrational packing. Granular Matter 21 (2), 1–16.

Darmana, D., Deen, N. G., Kuipers, J., 2006. Parallelization of an Euler–Lagrange model using mixed domain decomposition and a mirror domain technique: Application to dispersed gas–liquid two-phase flow. Journal of Computational Physics 220 (1), 216–248.

Dufresne, Y., Moureau, V., Lartigue, G., Simonin, O., 2020. A massively parallel CFD/DEM approach for reactive gas-solid flows in complex geometries using unstructured meshes. Computers & Fluids 198, 104402.

Durán, O., Claudin, P., Andreotti, B., 2011. On aeolian transport: Grain-scale interactions, dynamical mechanisms and scaling laws. Aeolian Research 3 (3), 243–270.

Gondret, P., Lance, M., Petit, L., 2002. Bouncing motion of spherical particles in fluids. Physics of Fluids 14 (2), 643–652.
Gopalakrishnan, P., Tafti, D., 2013. Development of parallel DEM for the open source code MFIX. Powder Technology 235, 33–41.

Graf, W. H., 1984. Hydraulics of sediment transport. Water Resources Publication, Colorado.

Hertz, H., 1882. Ueber die berührung fester elastischer körper. Journal für die reine und angewandte Mathematik 1882 (92), 156–171.

Ho, T. D., Valance, A., Dupont, P., El Moctar, A. O., 2011. Scaling laws in aeolian sand transport. Physical Review Letters 106 (9), 094501.

Huang, W.-X., Tian, F.-B., 2019. Recent trends and progress in the immersed boundary method. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 233 (23-24), 7617–7636.

Jain, R., Tschisgale, S., Fröhlich, J., 2021. Impact of shape: DNS of sediment transport with non-spherical particles. Journal of Fluid Mechanics 916, A38.

Ji, C., Munjiza, A., Avital, E., Xu, D., Williams, J., 2014. Saltation of particles in turbulent channel flow. Physical Review E 89 (5), 052202.

Jiménez, J., 2004. Turbulent flows over rough walls. Annual Review of Fluid Mechanics 36, 173–196.

Joseph, G., Hunt, M., 2004. Oblique particle-wall collisions in a liquid. Journal of Fluid Mechanics 510, 71–93.

Kadivar, M., Tormey, D., McGranaghan, G., 2021. A review on turbulent flow over rough surfaces: Fundamentals and theories. International Journal of Thermofluids 10, 100077.

Kafui, D., Johnson, S., Thornton, C., Seville, J., 2011. Parallelization of a Lagrangian–Eulerian DEM/CFD code for application to fluidized beds. Powder Technology 207 (1-3), 270–278.

Kempe, T., Fröhlich, J., 2012a. Collision modelling for the interface-resolved simulation of spherical particles in viscous fluids. Journal of Fluid Mechanics 709, 445–489.
Kempe, T., Fröhlich, J., 2012b. An improved immersed boundary method with direct forcing for the simulation of particle laden flows. Journal of Computational Physics 231 (9), 3663–3684.

Kempe, T., Vowinckel, B., Fröhlich, J., 2014. On the relevance of collision modeling for interface-resolving simulations of sediment transport in open channel flow. International Journal of Multiphase Flow 58, 214–235.

Kidanemariam, A. G., Uhlmann, M., 2014a. Direct numerical simulation of pattern formation in subaqueous sediment. Journal of Fluid Mechanics 750, R2.

Kidanemariam, A. G., Uhlmann, M., 2014b. Interface-resolved direct numerical simulation of the erosion of a sediment bed sheared by laminar channel flow. International Journal of Multiphase Flow 67, 174–188.

Kidanemariam, A. G., Uhlmann, M., 2017. Formation of sediment patterns in channel flow: minimal unstable systems and their temporal evolution. Journal of Fluid Mechanics 818, 716–743.

Kim, J., Moin, P., 1985. Application of a fractional-step method to incompressible Navier-Stokes equations. Journal of Computational Physics 59 (2), 308–323.

Kok, J. F., Parteli, E. J., Michaels, T. I., Karam, D. B., 2012. The physics of wind-blown sand and dust. Reports on Progress in Physics 75 (10), 106901.

Lanigan, D., Stout, J., Anderson, W., 2016. Atmospheric stability and diurnal patterns of aeolian saltation on the Llano Estacado. Aeolian Research 21, 131–137.

Le Roux, J. P., Rojas, E. M., 2007. Sediment transport patterns determined from grain size parameters: Overview and state of the art. Sedimentary Geology 202 (3), 473–488.

Luo, K., Wang, Z., Fan, J., Cen, K., 2007. Full-scale solutions to particle-laden flows: Multidirect forcing and immersed boundary method. Physical Review E 76 (6), 066709.

Merritt, W. S., Letcher, R. A., Jakeman, A. J., 2003. A review of erosion and sediment transport models. Environmental Modelling & Software 18 (8-9), 761–799.
Mittal, R., Iaccarino, G., 2005. Immersed boundary methods. Annual Review of Fluid Mechanics 37, 239–261.

Mordant, N., Pinton, J.-F., 2000. Velocity measurement of a settling sphere. The European Physical Journal B-Condensed Matter and Complex Systems 18 (2), 343–352.

Moureau, V., Domingo, P., Vervisch, L., 2011. Design of a massively parallel CFD code for complex geometries. Comptes Rendus Mécanique 339 (2-3), 141–148.

Munjiza, A., Andrews, K. R. F., 2000. Penalty function method for combined finite–discrete element systems comprising large number of separate bodies. International Journal for Numerical Methods in Engineering 49 (11), 1377–1396.

Munjiza, A., Owen, D. R. J., Bicanic, N., 1995. A combined finite-discrete element method in transient dynamics of fracturing solids. Engineering Computations 12 (2), 145–174.

Pähtz, T., Clark, A. H., Valyrakis, M., Durán, O., 2020. The physics of sediment transport initiation, cessation, and entrainment across aeolian and fluvial environments. Reviews of Geophysics 58 (1), e2019RG000679.

Papanicolaou, A. T. N., Elhakeem, M., Krallis, G., Prakash, S., Edinger, J., 2008. Sediment transport modeling review - current and future developments. Journal of Hydraulic Engineering 134 (1), 1–14.

Peng, C., Ayala, O. M., Wang, L.-P., 2019. A direct numerical investigation of two-way interactions in a particle-laden turbulent channel flow. Journal of Fluid Mechanics 875, 1096–1144.

Picano, F., Breugem, W.-P., Brandt, L., 2015. Turbulent channel flow of dense suspensions of neutrally buoyant spheres. Journal of Fluid Mechanics 764, 463–487.

Pozzetti, G., Jasak, H., Besseron, X., Roussel, A., Peters, B., 2019. A parallel dual-grid multiscale approach to CFD–DEM couplings. Journal of Computational Physics 378, 708–722.
Rana, S., Anderson, W., Day, M., 2021. An entrainment paradox: how hysteretic saltation and secondary transport augment atmospheric uptake of aeolian source materials. Journal of Geophysical Research: Atmospheres 126 (10), e2020JD033493.

Raupach, M. R., Antonia, R. A., Rajagopalan, S., 1991. Rough-wall turbulent boundary layers. Applied Mechanics Review 44 (1), 1–25.

Ray, S., Kempe, T., Fröhlich, J., 2015. Efficient modelling of particle collisions using a non-linear viscoelastic contact force. International Journal of Multiphase Flow 76, 101–110.

Roma, A. M., Peskin, C. S., Berger, M. J., 1999. An adaptive version of the immersed boundary method. Journal of Computational Physics 153 (2), 509–534.

Schiller, L., Naumann, A., 1933. Über die grundlegenden berechnungen bei der schwerkraftaufbereitung. Z. Vereines Deutscher Inge. 77, 318–321.

Shao, Y., 2008. Physics and modelling of wind erosion. Springer, Berlin.

Singh, K., Sandham, N., Williams, J., 2007. Numerical simulation of flow over a rough bed. Journal of Hydraulic Engineering 133 (4), 386–398.

Tao, S., Zhang, H., Guo, Z., Wang, L.-P., 2018. A combined immersed boundary and discrete unified gas kinetic scheme for particle–fluid flows. Journal of Computational Physics 375, 498–518.

Tenneti, S., Subramaniam, S., 2014. Particle-resolved direct numerical simulation for gas-solid flow model development. Annual Review of Fluid Mechanics 46, 199–230.

Tschisgale, S., Kempe, T., Fröhlich, J., 2017. A non-iterative immersed boundary method for spherical particles of arbitrary density ratio. Journal of Computational Physics 339, 432–452.

Tsuji, T., Yabumoto, K., Tanaka, T., 2008. Spontaneous structures in threedimensional bubbling gas-fluidized bed by parallel DEM–CFD coupling simulation. Powder Technology 184 (2), 132–140.
Uhlmann, M., 2004. Simulation of particulate flows multi-processor machines with distributed memory. Tech. Rep. CIEMAT-1039, Centro de Investigaciones Energeticas, Medioambientales y Tecnologicas (CIEMAT), Madrid (Spain).

Uhlmann, M., 2005. An immersed boundary method with direct forcing for the simulation of particulate flows. Journal of Computational Physics 209 (2), 448–476.

Valance, A., Rasmussen, K. R., El Moctar, A. O., Dupont, P., 2015. The physics of Aeolian sand transport. Comptes Rendus Physique 16 (1), 105–117.

Vowinckel, B., Kempe, T., Fröhlich, J., 2014. Fluid–particle interaction in turbulent open channel flow with fully-resolved mobile beds. Advances in Water Resources 72, 32–44.

Wang, G., Abbas, M., Climent, É., 2017a. Modulation of large-scale structures by neutrally buoyant and inertial finite-size particles in turbulent Couette flow. Physical Review Fluids 2 (8), 084302.

Wang, S., Luo, K., Yang, S., Hu, C., Fan, J., 2017b. Parallel LES–DEM simulation of dense flows in fluidized beds. Applied Thermal Engineering 111, 1523–1535.

Wang, S., Vanella, M., Balaras, E., 2019. A hydrodynamic stress model for simulating turbulence/particle interactions with immersed boundary methods. Journal of Computational Physics 382, 240–263.

Wiberg, P. L., Dungan Smith, J., 1989. Model for calculating bed load transport of sediment. Journal of Hydraulic Engineering 115 (1), 101–123.

Yang, Z., Deng, B.-Q., Shen, L., 2018. Direct numerical simulation of wind turbulence over breaking waves. Journal of Fluid Mechanics 850, 120–155.

Yang, Z., Lu, X.-H., Guo, X., Liu, Y., Shen, L., 2017. Numerical simulation of sediment suspension and transport under plunging breaking waves. Computers & Fluids 158, 57–71.

Yu, Z., Shao, X., 2007. A direct-forcing fictitious domain method for particulate flows. Journal of Computational Physics 227 (1), 292–314.
Zhang, Y., Hu, R., Zheng, X., 2018. Large-scale coherent structures of suspended dust concentration in the neutral atmospheric surface layer: A large-eddy simulation study. Physics of Fluids 30 (4), 046601.

Zheng, X., 2009. Mechanics of wind-blown sand movements. Springer, Berlin.

Zheng, X., Feng, S., Wang, P., 2021a. Modulation of turbulence by saltating particles on erodible bed surface. Journal of Fluid Mechanics 918, A16.

Zheng, X., Wang, G., Zhu, W., 2021b. Experimental study on the effects of particle–wall interactions on VLSM in sand-laden flows. Journal of Fluid Mechanics 914, A35.

Zhou, Q., Fan, L.-S., 2014. A second-order accurate immersed boundary-lattice Boltzmann method for particle-laden flows. Journal of Computational Physics 268, 269–301.

Zhu, H., Pan, C., Wang, G., Liang, Y., Ji, X., Wang, J., 2021. Attached eddy-like particle clustering in a turbulent boundary layer under net sedimentation conditions. Journal of Fluid Mechanics 920, A53.

Zhu, H.-Y., Pan, C., Wang, J.-J., Liang, Y.-R., Ji, X.-C., 2019. Sand-turbulence interaction in a high-Reynolds-number turbulent boundary layer under net sedimentation conditions. International Journal of Multiphase Flow 119, 56–71.