Research paper

SediFoam: A general-purpose, open-source CFD–DEM solver for particle-laden flow with emphasis on sediment transport

CrossMark

Rui Sun, Heng Xiao *

Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24060, United States

A B S T R A C T

With the growth of available computational resource, CFD–DEM (computational fluid dynamics–discrete element method) becomes an increasingly promising and feasible approach for the study of sediment transport. Several existing CFD–DEM solvers are applied in chemical engineering and mining industry. However, a robust CFD–DEM solver for the simulation of sediment transport is still desirable. In this work, the development of a three-dimensional, massively parallel, and open-source CFD–DEM solver *SediFoam* is detailed. This solver is built based on open-source solvers OpenFOAM and LAMMPS. OpenFOAM is a CFD toolbox that can perform three-dimensional fluid flow simulations on unstructured meshes; LAMMPS is a massively parallel DEM solver for molecular dynamics. Several validation tests of *SediFoam* are performed using cases of a wide range of complexities. The results obtained in the present simulations are consistent with those in the literature, which demonstrates the capability of *SediFoam* for sediment transport applications. In addition to the validation test, the parallel efficiency of *SediFoam* is studied to test the performance of the code for large-scale and complex simulations. The parallel efficiency tests show that the scalability of *SediFoam* is satisfactory in the simulations using up to $O(10^7)$ particles.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Particle-laden flows are frequently encountered in engineering applications such as coastal sediment transport, gas–solid fluidization, and aerosol deposition. Numerical simulations of these systems can improve the physical understanding of these flows. With the development of available computational resources, the CFD–DEM approach becomes an increasingly promising approach for particle-laden flows. In CFD–DEM, DEM approach tracks the motion of Lagrangian particles based on Newton's second law; CFD solves the motion of fluid flow based on locally averaged Navier–Stokes equations (Anderson and Jackson, 1967).

The CFD–DEM solvers include commercial solvers, research codes, and open-source solvers. These solvers solve similar equations for fluid flow and particle motion, and use similar submodels (i.e., drag force model and particle collision model). Commercial solvers, such as Fluent EDEM and the CFD–DEM packages in STAR-CCM+ and AVL-Fire (Spogis, 2008; Fries et al., 2011; Ebrahimi, 2014; Eppinger et al., 2011; Jajcevic et al., 2013), are general-purposed solvers without emphasis on sediment transport. In-house research codes (Calantoni et al., 2004; Deb and Tafti, 2013; Capecelatro and Desjardins, 2013; Wu et al., 2014)

can also be applied to CFD–DEM simulations, but the accessibility of these solvers is limited. On the other hand, open-source solvers (Garg et al., 2012; Li et al., 2012; Goniva et al., 2009) provide the user with good versatility in the development of the numerical model.

The efforts in the development of CFD–DEM solver focus on gas–solid flows. However, several unique features in sediment transport should be accounted for. First, for subaqueous sediment transport, the lubrication and the added mass force on the particle are much larger than gas–solid flows. The influences of them should be accounted for in CFD–DEM simulations. Moreover, sediment transport occurs at the boundary layer. To resolve the fluid flow, the size of CFD mesh can be smaller than sediment particle. Therefore, a robust algorithm is required when converting the properties of discrete particles to the Eulerian CFD mesh. Finally, the parallel efficiency of the code is important since the number of sediment particles can be as large as $O(10^6)$ in the simulation of laboratory-scale problems. Because these unique features of sediment transport are ignored in previous numerical simulations (Drake and Calantoni, 2001; Calantoni et al., 2004; Durán et al., 2012; Furbish and Schmeeckle, 2013; Schmeeckle, 2014), a robust open-source solver is still desirable for the study of sediment transport. In this work, a three-dimensional, massively parallel, and open-source CFD–DEM solver *SediFoam* with emphasis on sediment transport is presented. The originality of *SediFoam* includes (a) the lubrication and the added mass force on the particle; (b) the averaging algorithm to map the

properties of Lagrangian particles to Eulerian mesh (Sun and Xiao, 2015b); (c) the parallel algorithm and the performance test. The potential advantage of *SediFoam* includes the ability to simulate polydispersed particle mixtures, and the flexibility of applying turbulence models because of the robustness of OpenFOAM, among others (Wang et al., 2014). The present solver is developed based on two state-of-the-art open-source solvers, i.e., CFD solver OpenFOAM (Open Field Operation and Manipulation), and molecular dynamics simulator LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator). The two solvers are selected because they are both open-source, parallelized, highly modular and well established (OpenCFD, 2013; Plimpton, 1995).

The rest of this paper is organized as follows. The methodology of the present code is introduced in Section 2, including the mathematical formulation of fluid equations, particle motion equations, and fluid–particle interactions. Section 3 describes the implementations, including the communication and the parallelization of the code. The numerical validations of the code for various sediment transport problems are performed in Section 4. The parallel efficiency of the present solver is detailed in Section 5. Finally, Section 6 concludes the paper.

2. Methodology

2.1. Mathematical model of particle motion

In the CFD–DEM approach, the translational and rotational motion of each particle is calculated based on Newton's second law as the following equations (Cundall and Strack, 1979; Ball and Melrose, 1997):

$$m\frac{d\mathbf{u}}{dt} = \mathbf{f}^{col} + \mathbf{f}^{lub} + \mathbf{f}^{fp} + m\mathbf{g}, \tag{1a}$$

$$I\frac{d\mathbf{\Psi}}{dt} = \mathbf{T}^{col} + \mathbf{T}^{lub} + \mathbf{T}^{fp}, \tag{1b}$$

where $\mathbf{u}$ is the velocity of the particle; t is the time; m is the particle mass; $\mathbf{f}^{col}$ represents the contact forces due to particle–particle or particle–wall collisions; $\mathbf{f}^{lub}$ is the lubrication force due to the fluid squeezed out from the gaps between two particles; $\mathbf{f}^{fp}$ denotes fluid–particle interaction forces (e.g., drag, lift force, added mass force, and buoyancy); $\mathbf{g}$ denotes body force. Similarly, I and $\mathbf{\Psi}$ are angular moment of inertia and angular velocity, respectively, of the particle; $\mathbf{T}^{col}$, $\mathbf{T}^{lub}$ and $\mathbf{T}^{fp}$ are the torques due to contact forces, lubrication, and fluid–particle interactions, respectively. To compute the collision forces and torques, the particles are modeled as soft spheres with inter-particle contact represented by an elastic spring and a viscous dashpot. Further details can be found in Tsuji et al. (1993).

2.2. Locally averaged Navier–Stokes equations for fluids

The fluid phase is described by the locally averaged incompressible Navier–Stokes equations. Assuming constant fluid density ρ_f, the governing equations for the fluid are (Anderson and Jackson, 1967; Kafui et al., 2002):

$$\nabla \cdot \left(\varepsilon_s \mathbf{U}_s + \varepsilon_f \mathbf{U}_f \right) = 0, \tag{2a}$$

$$\frac{\partial \left(\varepsilon_f \mathbf{U}_f \right)}{\partial t} + \nabla \cdot \left(\varepsilon_f \mathbf{U}_f \mathbf{U}_f \right) = \frac{1}{\rho_f} \left(-\nabla p + \varepsilon_f \nabla \cdot \mathcal{R} + \varepsilon_f \rho_f \mathbf{g} + \mathbf{F}^{fp} \right), \tag{2b}$$

where ε_s is the solid volume fraction; $\varepsilon_f = 1 - \varepsilon_s$ is the fluid volume fraction; $\mathbf{U}_f$ is the fluid velocity. The terms on the right-hand side of the momentum equation are pressure (p) gradient, divergence of the stress tensor $\mathcal{R}$, gravity, and fluid–particle interactions forces, respectively. Large eddy simulation is performed in the present work, the stress tensor is composed of both viscous and Reynolds stresses $\mathcal{R} = \mu \nabla \mathbf{U}_f + \rho_f \tau$, where μ is the dynamic viscosity of the fluid flow; τ is the Reynolds stress. The expression of the Reynolds stress is:

$$\tau = \frac{2}{\rho_f} \mu_t \mathbf{S} - \frac{2}{3} k \mathbf{I}; \tag{3}$$

where μ_t is the dynamics eddy viscosity; $\mathbf{S} = (\nabla \mathbf{U}_f + (\nabla \mathbf{U}_f)^T)/2$; k is the turbulent kinetic energy. It is noted that in the stress tensor $\mathcal{R}$ term, the fluctuations of the fluid flow at the boundary of the particle are not resolved. The Eulerian fields ε_s, $\mathbf{U}_s$, and $\mathbf{F}^{fp}$ in Eq. (2) are obtained by averaging the information of Lagrangian particles. However, the fluid variables are not averaged before sending to the particle, which is because the averaged fluid variables might be too diffusive for calculation. It is noted that the coupling of the fluid and solid phases is similar with Scheme 3 according to the literature (Feng and Yu, 2004; Zhou et al., 2010). At each time step, the fluid–particle interaction forces on individual particles are calculated first in the CFD cell, and the values are then summed to produce the interaction force at the cell scale. Detailed information of the coupling between the solvers can be found in Section 3.

2.3. Fluid–particle interactions

The fluid–particle interaction force $\mathbf{F}^{fp}$ consists of buoyancy $\mathbf{F}^{buoy}$, drag $\mathbf{F}^{drag}$, lift force $\mathbf{F}^{lift}$, and added mass force $\mathbf{F}^{add}$. Although the lift force and the added mass force are usually ignored in fluidized bed simulation, they are important in the simulation of sediment transport.

The drag on an individual particle i is formulated as:

$$\mathbf{f}_i^{drag} = \frac{V_{p,i}}{\varepsilon_{f,i} \varepsilon_{s,i}} \beta_i \left(\mathbf{u}_{p,i} - \mathbf{U}_{f,i} \right), \tag{4}$$

where $V_{p,i}$ and $\mathbf{u}_{p,i}$ are the volume and the velocity of particle i, respectively; $\mathbf{U}_{f,i}$ is the fluid velocity interpolated to the center of particle i; β_i is the drag correlation coefficient which accounts for the presence of other particles. The β_i value for the drag force is based on Syamlal et al. (1993):

$$\beta_i = \frac{3}{4} \frac{C_d}{V_r^2} \frac{\rho_f \left| \mathbf{u}_{p,i} - \mathbf{U}_{f,i} \right|}{d_{p,i}}, \quad \text{with } C_d = \left(0.63 + 0.48 \sqrt{V_r/Re} \right), \tag{5}$$

where the particle Reynolds number Re is defined as:

$$Re = \rho_i d_{p,i} \left| \mathbf{u}_{p,i} - \mathbf{U}_{f,i} \right|; \tag{6}$$

the V_r is the correlation term:

$$V_r = 0.5 \left(A_1 - 0.06Re + \sqrt{(0.06Re)^2 + 0.12Re(2A_2 - A_1) + A_1^2} \right), \tag{7}$$

with

$$A_1 = \varepsilon_f^{4.14}, \quad A_2 = \begin{cases} 0.8\varepsilon_f^{1.28} & \text{if } \varepsilon_f \leq 0.85, \\ \varepsilon_f^{2.65} & \text{if } \varepsilon_f > 0.85. \end{cases} \tag{8}$$

Note that other drag force models are also implemented in *Sedi-Foam* with correlations for β in dense particle-laden flows (Di Felice, 1994; Wen and Yu, 1966). In addition to drag, the lift force on a spherical particle is modeled in *SediFoam* as (Saffman, 1965;

van Rijn, 1984):

$$\mathbf{f}_i^{lift} = C_l \varrho_f \nu^{0.5} D^2 \left(\mathbf{u}_{p,i} - \mathbf{U}_{f,i} \right) \times \nabla \mathbf{U}_{f,i}, \tag{9}$$

where $\times$ indicates the cross product of two vectors, and $C_l = 1.6$ is the lift coefficient. The added mass force is considered in *SediFoam* due to the comparable densities of the carrier and disperse phases in sediment transport applications. This is modeled as:

$$\mathbf{f}_i^{add} = C_{add} \varrho_f V_{p,i} \left(\frac{D\mathbf{u}_{p,i}}{Dt} - \frac{D\mathbf{U}_{f,i}}{Dt} \right), \tag{10}$$

where $C_{add} = 0.5$ is the coefficient of added mass. The Lagrangian particle acceleration term $D\mathbf{u}_{p,i}/Dt$ utilizes the particle velocity at the previous time; the material derivative of fluid phase velocity $D\mathbf{U}_{f,i}/Dt = d\mathbf{U}_{f,i}/dt + \mathbf{U}_{f,i} \cdot \nabla \mathbf{U}_{f,i}$ can be obtained in OpenFOAM at each time step. The lift force and added mass models used in the present work are only applicable for particle-laden flows in small Reynolds number and low volume fraction regime. The study of van Rijn (1984) has indicated that the accuracy of such modeling scheme is acceptable for sediment transport applications.

2.4. Diffusion-based averaging method

According to Eq. (2), the continuum Eulerian fields of ε_s, $\mathbf{U}_s$, and $\mathbf{F}^{fp}$ are obtained by averaging from discrete particle data. In the present solver, the averaging algorithm previously proposed by the authors (Sin and Xiao, 2015a,b) is implemented.

Taking the averaging process of ε_s as an example. In the first step, the particle volumes at each CFD cell are obtained. Then, the solid volume fraction for cell k is calculated by dividing the total particle volume by the volume of this cell $V_{c,k}$. That is:

$$\varepsilon_{s,k}(\mathbf{x}, \tau = 0) = \frac{\sum_{i=1}^{n_{p,i}} V_{p,i}}{V_{c,k}}, \tag{11}$$

where $n_{p,k}$ is the number of particles in cell k. With the initial condition in Eq. (11), a transient diffusion equation for $\varepsilon_s(\mathbf{x}, \tau)$ is solved to obtain the continuum Eulerian field of ε_s:

$$\frac{\partial \varepsilon_s}{\partial \tau} = \nabla^2 \varepsilon_s \tag{12}$$

where ∇^2 is the Laplacian operator; τ is the pseudo-time. It has been established in Sin and Xiao (2015a,b) that the results obtained by Eq. (12) is equivalent with Gaussian kernel based averaging with bandwidth $b = \sqrt{4\tau}$. Similarly, the smoothed $\mathbf{U}_s$ and $\mathbf{F}^{fp}$ fields can be obtained by using this approach.

We note that a similar averaging procedure has been proposed earlier by Capecelatro and Desjardins (2013), where a mollification procedure with Gaussian kernel is followed by solving a diffusion equation of the obtained volume fraction. Both methods are conservative and mesh-independent, and are theoretically equivalent. The novelty of Sin and Xiao (2015a,b) lies in the following aspects. First, they established the theoretical equivalence between the diffusion based averaging procedure and the Gaussian kernel averaging commonly used in statistical mechanics (Zhu and Yu, 2002). Based on this insight, they provided a clear physical interpretation of the normalized diffusion time, rendering it a physical parameter related to the wake of the particles. Second, the wall boundaries are treated in Sun and Xiao (2015b) with a straightforward, efficient no-flux boundary condition, and the conservativeness has been proved analytically. In contrast, Capecelatro and Desjardins (2013) ensured the conservativeness of the averaging procedure at wall boundaries by using ghost particles. Finally, the coarse-graining procedure of Sin and Xiao (2015a,b) is implemented based on the open-source, general-purpose, three-dimensional, massively parallel CFD solver with a generic

unstructured body-fitting mesh, while Capecelatro and Desjardins (2013) used an in-house CFD solver based on a Cartesian mesh with an immersed boundary method. More details of the averaging algorithm can be found in Sin and Xiao (2015a,b).

2.5. Numerical methods

The solution of the particle motions including their interactions via collisions and endured contacts are handled by LAMMPS. The fluid forces $\mathbf{f}^{fp}$ on the particles are computed in OpenFOAM, supplied into LAMMPS, and used in the integration of particle motion equations (1). The particle forces in the fluid equations are computed in OpenFOAM according to the forces on individual particles via the averaging procedure.

The fluid equations in (2) are solved by OpenFOAM using the finite volume method (Jasak, 1996). The discretization is based on a collocated grid, i.e., pressure and all velocity components are stored in cell centers. PISO (Pressure Implicit Splitting Operation) algorithm is used for velocity–pressure decoupling (Issa, 1986). Second-order central schemes are used for the spatial discretization of convection terms and diffusion terms. Time integrations are performed using a second-order implicit scheme. The averaging method involves solving transient diffusion equations based on the OpenFOAM platform. The diffusion equations are also solved on the CFD mesh. A second-order central scheme is used for the spatial discretization of the diffusion equation; a second-order implicit scheme is used for the temporal integration.

To solve the equation of motion of the particles in (1), the nearest particles of each particle are tracked. To find the nearest particles, LAMMPS uses a combination of neighbor lists and link-cell binning (Hockney et al., 1974) and the scale of the computation is only $O(N)$ (Plimpton, 1995). The collision force is computed with a linear spring–dashpot model, in which the normal elastic contact force between two particles is linearly proportional to the overlapping distance (Cundall and Strack, 1979). The lubrication model is based on Ball and Melrose (1997), in which the force is proportional to the relative velocity and inversely proportional to the relative distance. The torque on the particles due to the lubrication and collision are integrated to calculate the particle rotation, but the interaction of the particle rotation and fluid is not considered. Finally, the time step to resolve the particle collision is 1/50 the contact time to avoid particle inter-penetration (Sun et al., 2007).

3. Implementations

SediFoam was originally developed by the second author and his co-workers to study particle segregation dynamics (Sun et al., 2009). To improve the solver, we enhanced its parallel computing capabilities and implemented the averaging algorithm for sediment transport. The source code of *SediFoam* is available at https://github.com/xiaoh/sediFoam.

The diagram of the code is shown in Fig. 1. It can be seen in the figure that the fluid and particle equations are solved individually by the CFD and DEM modules at each time step. The averaging procedure is performed in the CFD module before solving the fluid equations. In addition to solving the equations, the information of the sediment particles is updated before CFD module starts the averaging procedure; the fluid–particle interaction force of each particle is updated before DEM module evolves the motion of the particles. These procedures before solving the fluid and particle equations are the coupling between CFD and DEM modules.

In the parallelization of *SediFoam*, which is essential for simulating large granular systems, the equations are solved in parallel by the DEM and CFD modules. Therefore, the parallelization of
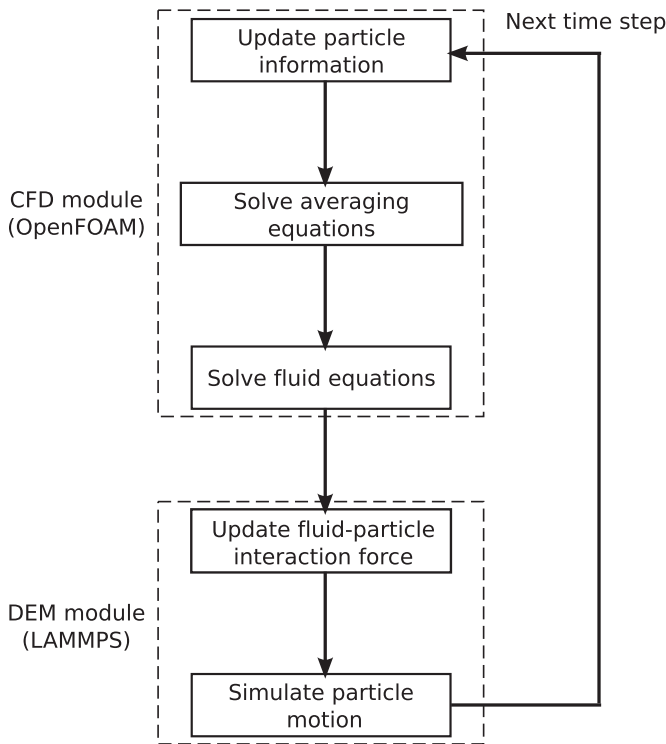
Fig. 1. The block diagram of the code.

Fig. 3. The geometry of a representative CFD–DEM case that the decompositions of the CFD and the DEM modules are different. The blue lines are the CFD mesh and the black dash lines illustrate the subdomains of the CFD module; the red lines are the subdomains of the DEM module. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

SediFoam concerns with the coupling of the modules between multiple processors. When using multiple processors to accelerate the simulation, both modules decompose the computational domain into N_{proc} subdomains. In the coupling procedure, the particle information obtained in each module is transferred to the other module. If the subdomains of individual processors in each module were perfectly consistent, the information of every particle would be local to each processor for both CFD and DEM modules. In this situation, inter-processor communication is unnecessary in the coupling. However, the subdomains in most numerical simulations are inconsistent (see Fig. 2). This is due to the consideration of parallel efficiency when decomposing the domain. In this situation, the information of some particles obtained
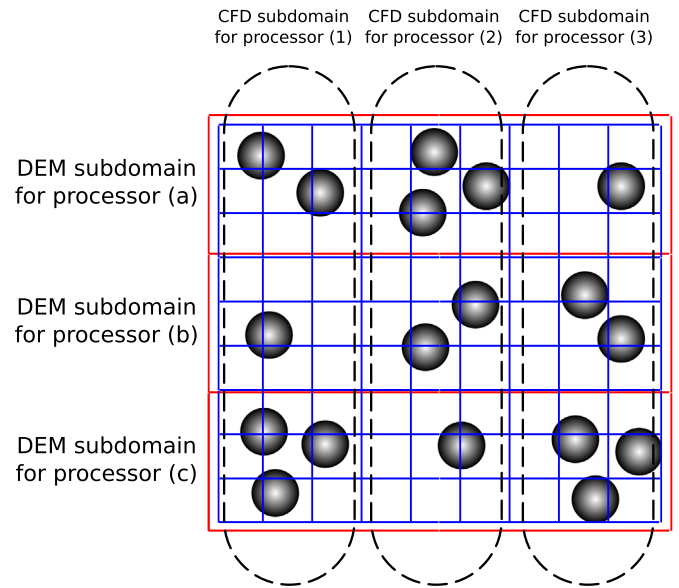
in CFD and DEM module are not stored in the same processor. Hence, inter-processor communication is required for these non-local particles.

An example is used to describe the parallelization of the coupling procedure in *SediFoam*. The geometry of this example employing three processors is shown in Fig. 3. The parallelization in the present solver is performed as follows:

1. The DEM module of *SediFoam* evolved the particles one step forward, shown in Fig. 4(a).
2. The non-local particles are found in the DEM module, shown in Fig. 4(b). This is the preparation step before transferring non-local data.
3. In each processor, the information of non-local particles is transferred to other processors. This step is illustrated in Fig. 4 (c).
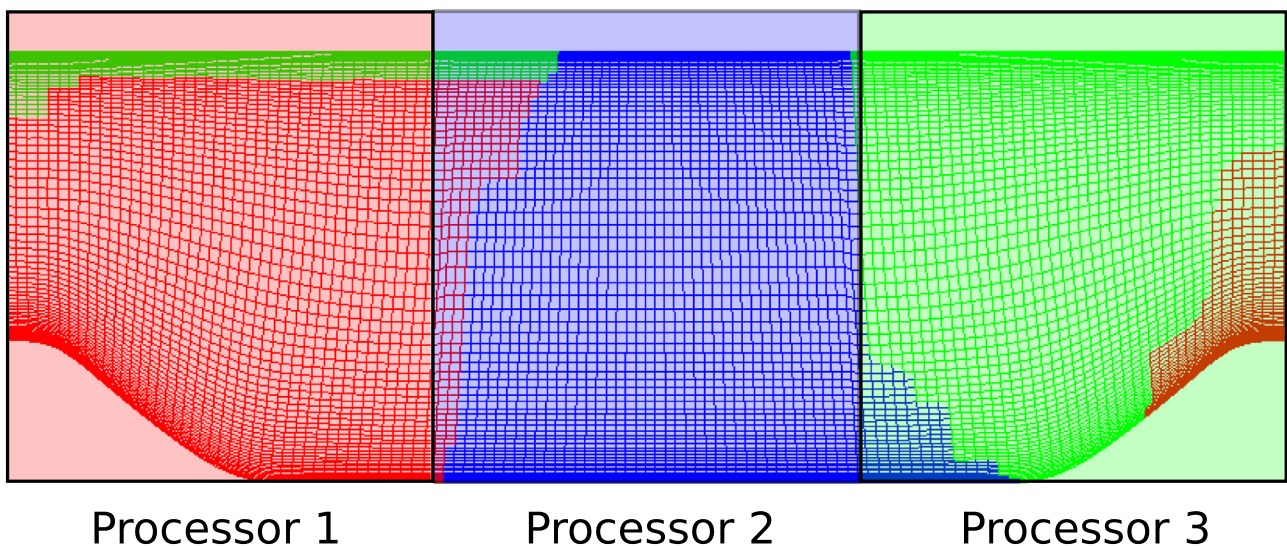

Fig. 2. Decomposition of the computational domain in CFD–DEM simulation. Different colors in the CFD mesh denote different subdomains in the CFD module, while the bound boxes of different colors denote different subdomains in the DEM module. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)
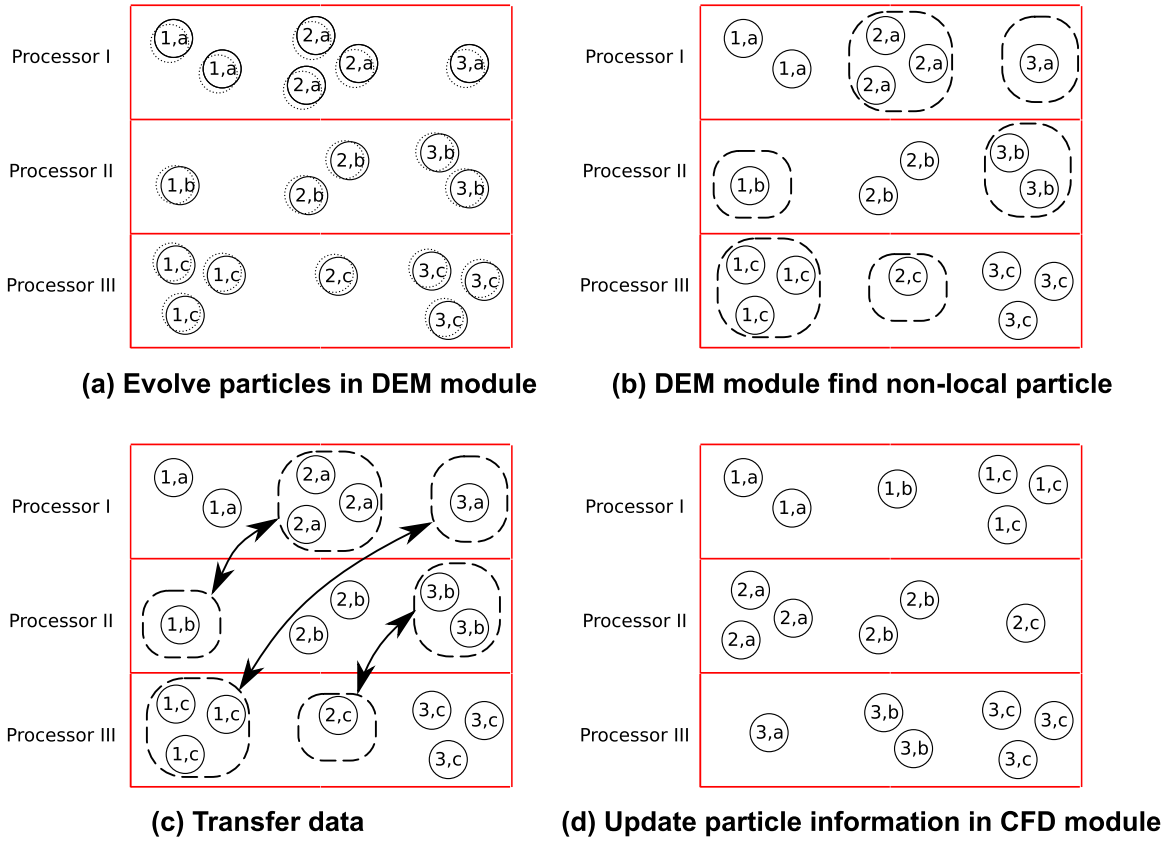
(a) Evolve particles in DEM module

(b) DEM module find non-local particle

(c) Transfer data

(d) Update particle information in CFD module

Fig. 4. Inter-processor communication of the particle information. Each circle represents a sediment particle in the simulation. Numbers 1, 2, and 3 indicate the processor that stores the data of the particle in the CFD module; characters "a", "b" and "c" denote the processor that store the data of the particle in the DEM module.

4. The particle information obtained in the DEM module is now local to the CFD module, as shown in Fig. 4(d). The information of the particles in the CFD module is updated and can be used in the next CFD step.

After Step 4, the inter-processor communication is finished. The non-local data obtained by the DEM module is transferred to update the information of sediment particles in the CFD module. Following this approach, the non-local information obtained in the CFD module can also be transferred to the particles in the DEM module.

4. Results

Extensive validation tests of *SediFoam* have been performed previously for fluidized bed simulations (Sun et al., 2009; Xiao and Sun, 2011; Gupta, 2015; Sun and Xiao, 2015a). In the present work, three numerical tests are presented to demonstrate the capability of *SediFoam* in the simulation of sediment transport. The sedimentation of a single particle in water is detailed in Section 4.1, which aims to validate the implementation of lubrication and added mass. The motion of 500 particles on fixed sediment bed is discussed in Section 4.2. The purpose of this case is to validate the properties of the fluid and sediment particles obtained by using *SediFoam*. Simulations of relatively large number of particles ($O(10^5)$) are detailed in Section 4.3. The objective of this test is to demonstrate the capability of the present solver in the simulation of large and complex cases.

4.1. Case 1: single particle sedimentation in water

Most numerical simulations using CFD–DEM are performed to study gas–solid flows. However, the behavior of a sediment particle in liquid is different from that in gaseous flow. This is because liquid has higher density and dynamic viscosity than gas. As such, lubrication and virtual mass force of the same particle in liquid–solid flows are approximately $O(10^3)$ times larger than those in gas–solid flows, and can be comparable to the weight of the sediment particle. Therefore, the influences of lubrication and virtual mass (usually negligible in gas–solid flows) can be critical in the simulation of subaqueous sediment transport. To test the implementation of the two forces in *SediFoam*, a series of simulations of particle–wall collision are performed based on the experiments of Gondret et al. (2002).

The geometry of the domain in the particle–wall collision test is shown in Fig. 5 along with the coordinate system. The parameters used in this case are detailed in Table 1. Periodic boundary condition is applied at the boundaries in both x- and z-directions. In the simulation, the fluid is quiescent initially and the particle falls in the vertical direction due to the gravity force. The initial particle–wall distance is large enough so that the particle accelerates to the terminal velocity before the collision occurs.

To test the influence of lubrication and added mass, the locations of the particle obtained in the collision test are compared with the experimental results. The comparison is performed at two different Stokes numbers, which is defined as:

$$St = \frac{\rho_p D_p u_p}{9\nu_f \rho_f} = \frac{\rho_p}{9\rho_f} Re_p.$$

(13)

In Fig. 6, the locations of the center of the sphere are plotted as a function of time. It can be seen that the results obtained in the simulation that considered both lubrication and added mass are consistent with the experimental measurements. Theoretically, added mass force adds to the inertia of the particle, which leads to
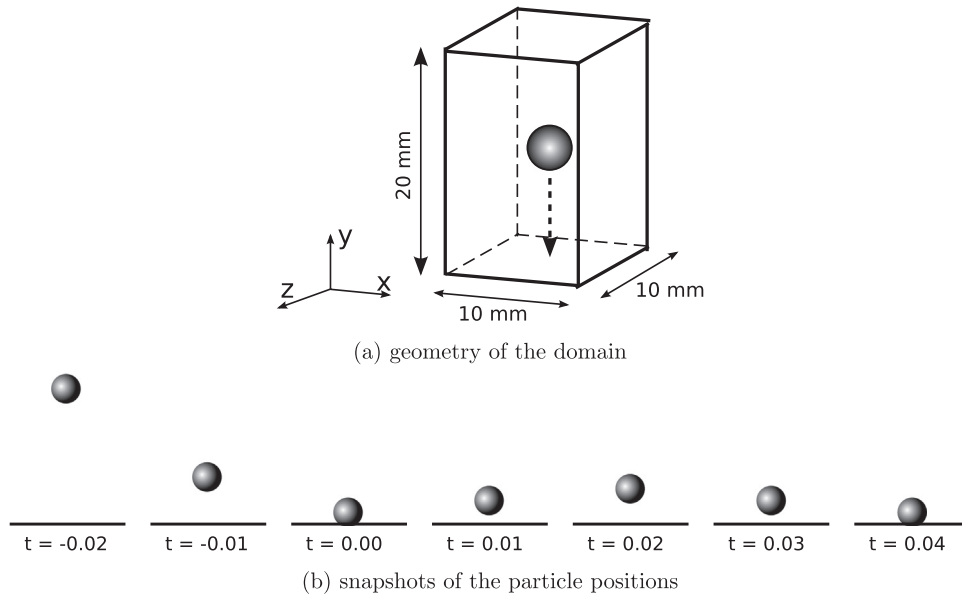
(a) geometry of the domain

| t = -0.02 | t = -0.01 | t = 0.00 | t = 0.01 | t = 0.02 | t = 0.03 | t = 0.04 |

(b) snapshots of the particle positions

Fig. 5. (a) Geometry of the domain and (b) snapshots of the positions of the particle in the collision test. The dash line in panel (a) illustrates the direction of particle motion. In panel (b), $t=0$ s responds to the moment the particle hits the wall for the first time.

Table 1
Parameters of the numerical simulations.

	Case 1	Case 2	Case 3
Domain dimensions			
Width (L_x) (mm)	100	58.4	120
Height (L_y) (mm)	200	13.5	40
Transverse thickness (L_z) (mm)	100	29.2	60
Mesh resolutions			
Width (N_x)	10	52	140
Height (N_y)	20	102	65
Transverse thickness (N_z)	10	26	60
Particle properties			
Total number	1	500	3.3×10^5
Diameter d_p (mm)	6	1.12	0.5
Density ρ_s ($\times 10^3$ kg/m^3)	2.5/7.8 (sand/steel)	2.0	2.5
Particle stiffness coefficient (N/m)	800	800	800
Normal restitution coefficient	0.97	0.97	0.01
Coefficient of friction	0.1	0.1	0.6
Fluid properties			
Viscosity ν ($\times 10^{-6}$ m^2/)	1.0–5.4	1.0	1.0
Density ρ_f ($\times 10^3$ kg/m^3)	1.0	1.0	1.0

larger rebound height regardless of the Stokes number. On the other hand, the lubrication depends on the viscous effect and decreases with Stokes number. At $St=27$, the viscous effect is large by definition. Consequently, the locations of the particle predicted without lubrication are significantly different from the experimental data. At $St=742$, the viscous effects are small, so accounting for lubrication at large Stokes number does not significantly influence the predictions of the particle motions.

The prediction of the effective restitution coefficient using *SediFoam* in particle–wall bouncing test is shown in Fig. 7. The effective restitution coefficient is defined as $e = e_{water}/e_{air}$, where e_{water} and e_{air} are the restitution coefficients for the same collision that occurs in water and air, respectively. The amount of decrease of e from 1 denotes the influence of lubrication and added mass in the collision. It can be seen in Fig. 7 that the predictions from *SediFoam* agree favorably with the experimental data when the influence of lubrication and added mass is considered. In contrast, the predictions without accounting for these forces deviate significantly from experimental measurements by over-predicting the effective restitution coefficients, particularly at small Stokes number. Accordingly, if sediment transport occurs at relatively low Stokes number (Kidanemariam and Uhlmann, 2014b), the
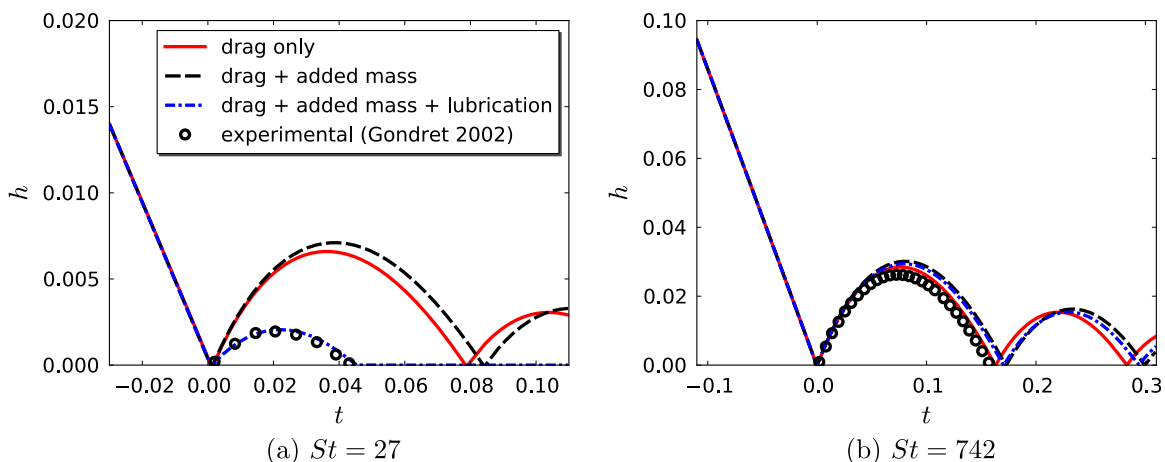

(a) $St = 27$

(b) $St = 742$

Fig. 6. The positions of the particle plotted as a function of time in the particle–wall bouncing test. The influence of lubrication and added mass are considered at Stokes number: (a) $St=27$ and (b) $St=742$.
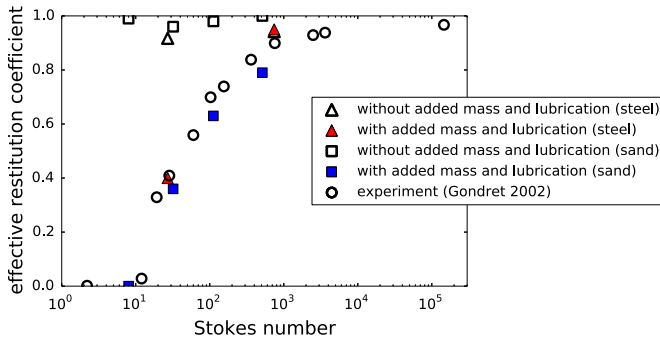
Fig. 7. The influence of lubrication and added mass on the effective restitution coefficient in particle–wall collisions for different particles at different Stokes numbers.

influence of lubrication and added mass should be considered in CFD–DEM modeling of sediment transport. Note that the calculation of lubrication incurs significant increase in computational costs. Consequently, lubrication are accounted for by using a smaller dry restitution coefficient in the literature (Schmeeckle, 2014; Kidanemariam and Uhlmann, 2014a).

In summary, the influence of added mass and lubrication is important in the CFD–DEM modeling of sediment transport. Additionally, the test cases validate the implementation of the added mass force and lubrication force in *SediFoam*.

4.2. Case 2: sediment transport with 500 particles

In this case, sediment transport at the boundary layer in the channel is studied. The results obtained by using *SediFoam* are compared with the numerical benchmark to illustrate the capability of the present solver. The averaging algorithm proposed by Sun and Xiao (2015b) is applied to obtain the continuum Eulerian quantities from discrete particles. This algorithm enables CFD–DEM simulation at the boundary layer where the size of CFD cells is smaller than particles.

The numerical setup is based on the numerical benchmark studying the motion of 500 movable particles (Kempe et al., 2014). The geometry of the simulation is shown in Fig. 8. The dimensions of the domain, the mesh resolutions, and the fluid and particle properties used are detailed in Table 1. The CFD mesh in the vertical (y-) direction is progressively refined towards the bottom boundary. Periodic boundary condition is applied in both x- and z-directions, no-slip wall condition is applied at the bottom in the y-direction, and slip wall condition is applied on the top in the y-direction. The bulk Reynolds number Re_b of the flow in the channel is 3010. Six layers of fixed particles are arranged hexagonally to provide a bottom boundary condition to the moving particles, as is shown in Fig. 8. In the coordinate system, the top of the fixed particle bed is at $y=0$. To obtain the profiles of fluid velocity and Reynolds stresses, the simulations are averaged for 50 flow-through times, and spatial average is performed in the horizontal domain. The bandwidth b used in the averaging procedure is $2d_p$
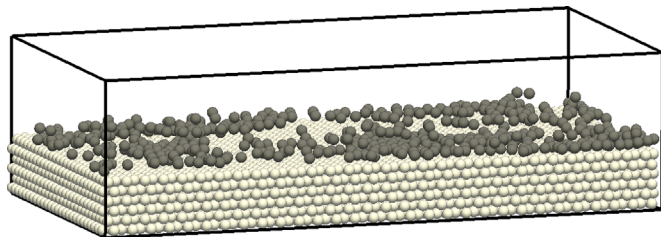

Fig. 8. Layout of sediment transport simulation according to Kempe et al. (2014). The white particles are fixed on the bottom; the gray particles are movable.

in width and thickness directions and d_p in vertical direction.

The flow properties are presented in Fig. 9, including the streamwise velocity and Reynolds stresses R_{xx}, R_{xy}, and R_{yy}. Fig. 9 (a) shows that *SediFoam* is able to capture the decrease of the fluid velocity in the near-wall boundary layer and within the bed, which is due to the drag force of the sediment particles. The velocity profile near the particle bed at $y=0$ is negative because of the diffusion effect of the averaging algorithm. Therefore, the diffusion bandwidth in the vertical direction is taken as small as d_p to reduce the effect of the numerical diffusion. It can be seen from Fig. 9(b)–(d) that the predictions of different components of Reynolds stresses from the present solver agree well with the numerical benchmark. Compared with the flow that has no particles in the channel, *SediFoam* captures the increase of the Reynolds stresses induced by the motion and collision of particles. It can be seen that the overall agreement of results obtained by using *SediFoam* and DNS is good for R_{xx} and R_{yy}. The discrepancy in R_{yy} is slightly larger than other components, which may be attributed to the fact that CFD–DEM do not resolve the flow fluctuation at the particle surface and cannot capture this quantity as good as the DNS.

Other quantities of interest in sediment transport include solid volume fraction and particle velocity. Fig. 10 demonstrates the time-averaged probability density function (PDF) and streamwise velocity of the particles at different vertical locations. It can be seen that both the probability density function and streamwise velocity agree well with the results in the numerical benchmark (Kempe et al., 2014). In the present simulation, the averaged streamwise velocity of all particles u_p/u_b is 0.28. Compared with the prediction 0.35 from the numerical benchmark, the error is 20%. However, this error is insignificant since u_p is proportional to the sediment transport rate, which varies significantly in the experimental measurements. Therefore, the accuracy of CFD–DEM modeling is acceptable. It is noteworthy considering the fact that the total computational costs of CFD–DEM are much smaller than the interface-resolved method. For this case, the number of CFD mesh in the present simulation is 1.4×10^5, whereas this number in the DNS simulation is 1.0×10^8.

4.3. Case 3: sediment transport with $O(10^5)$ particles

The purpose of this simulation is to test the performance of the present code in the simulation of larger problems of sediment transport. The results of both bed load sediment transport and suspended sediment transport are demonstrated.

The layout of this case is similar to Case 2. The domain geometry, the mesh resolution, and the properties of fluid and particles are detailed in Table 1. Periodic boundary condition is applied in both x- and z-directions. Slip wall condition is applied at the top of the domain, and no-slip wall condition is applied at the bottom. Three layers of solid particles are fixed at the bottom to provide rough wall boundary condition for DEM simulation. The flow velocities in five numerical simulations range from 0.3 m/s to 1.1 m/s. Each simulation is performed for 50 flow-through times for time averaging. The bandwidth b used in the averaging procedure is $4d_p$ in width and thickness directions and $2d_p$ in vertical direction. The geometry of this case is identical to the numerical tests using the same number of particles by Schmeeckle (2014).

The averaged properties of sediment particles are presented in Fig. 11, including the sediment transport rate and the friction coefficient. The sediment transport rate q_{sx} is obtained by multiplying the mean streamwise velocity of all particles by the total volume of the particles, and divide it by the area of the horizontal plane. The non-dimensional sediment fluxes $q_* = q_{sx}/((\rho_s/\rho_f - 1)gd_p^3)^{1/2}$ at different Shields parameters $\tau_* = \rho_f u_*^2/(\rho_s - \rho_f)gd_p$ are shown in Fig. 11(b). It can be seen in the figure
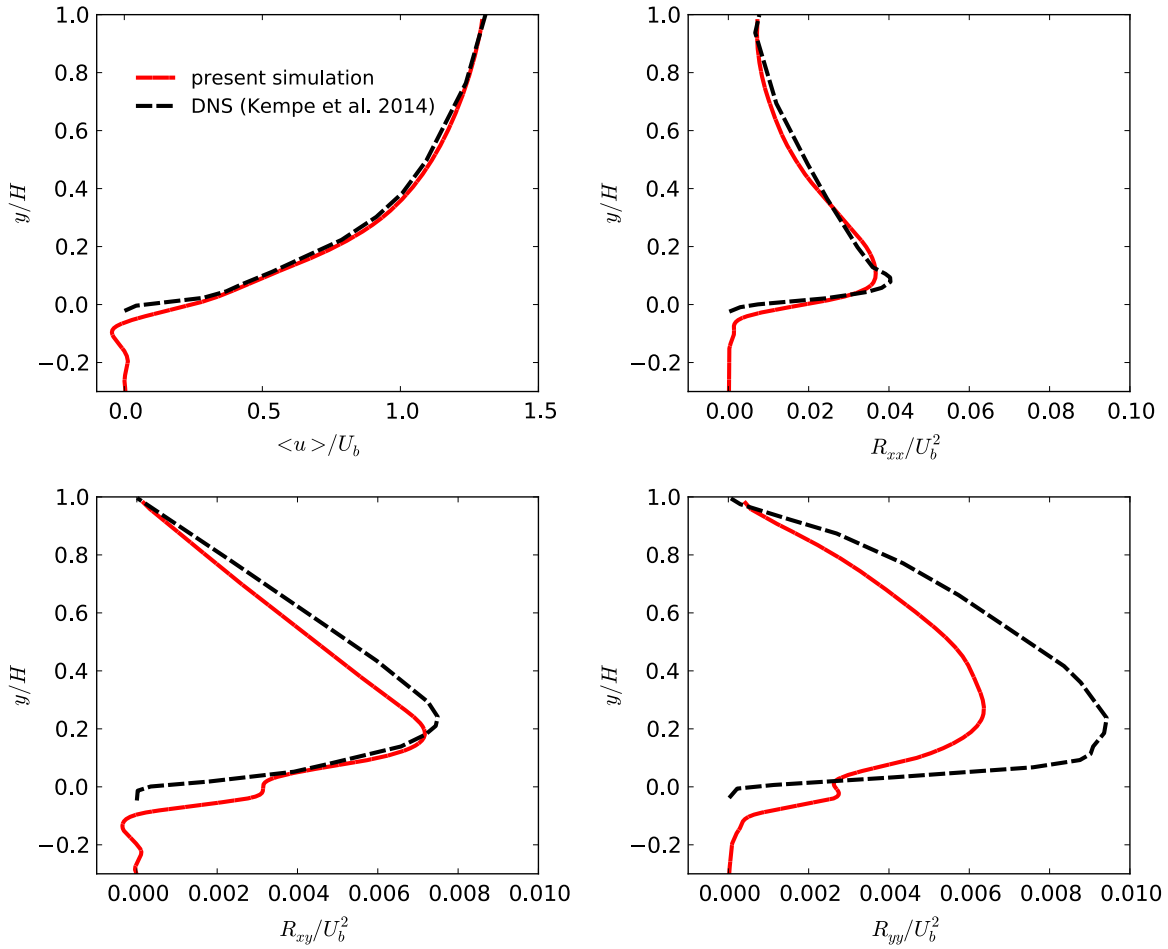
Fig. 9. Comparison of mean velocity $\langle u \rangle$ and Reynolds stresses R_{xx}, R_{xy}, and R_{yy} along the wall normal direction (y) obtained by using *SediFoam* with the DNS results (Kempe et al., 2014). The mean location of the particle bed is at $y = 0$.

that the sediment transport rates agree favorably with the experimental data. Note that q_* in different regimes (i.e., bed load and suspended load) used different regression curves. It is worth mentioning that the predictions of q_* in suspended load regime using *SediFoam* are better than the results obtained in the literature (Schmeeckle, 2014). This is because the drag force model in the present simulations can better predict the motion of sediment particles since the influence of ε_s is accounted for. The coefficient of friction of the surface is defined as $C_f = u_*^2 / \langle u \rangle^2$ and describes the

roughness of the bed. Shown in Fig. 11(c), C_f obtained in the present simulation are larger than the predictions by the law of the wall (i.e., Nikuradse value). This increase in C_f is because the hydraulic roughness over a loose bed is larger in the presence of movable particles. Note that the C_f predicted by *SediFoam* is slightly smaller than the results predicted by Schmeeckle (2014). This slight difference is because Schmeeckle (2014) over-estimated the volume fraction of the fluid and used a smaller $\langle u \rangle$ when calculating C_f.
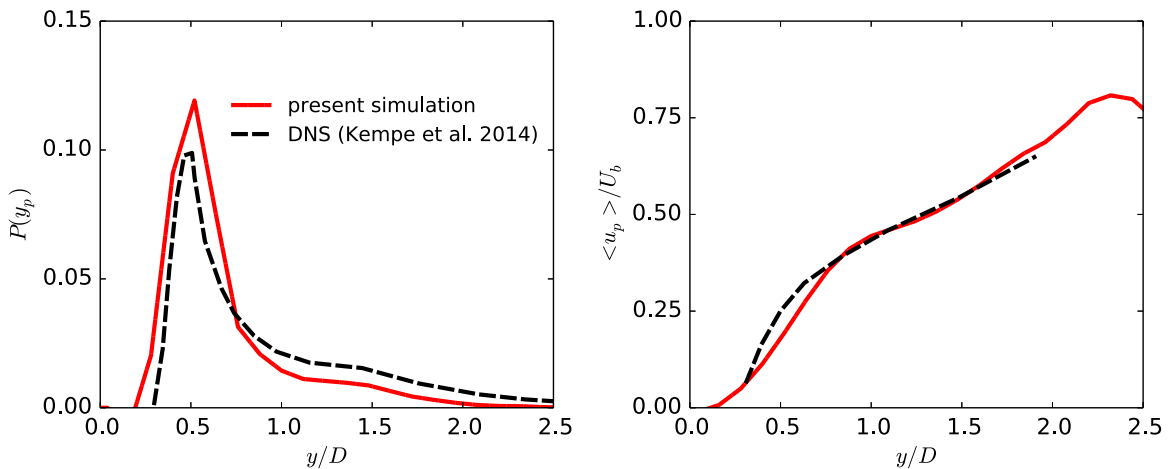

Fig. 10. Probability density function and streamwise velocity in the simulation of 500 movable particles.

(a) Distribution and velocity of sediment particles

(b) Sediment transport rate
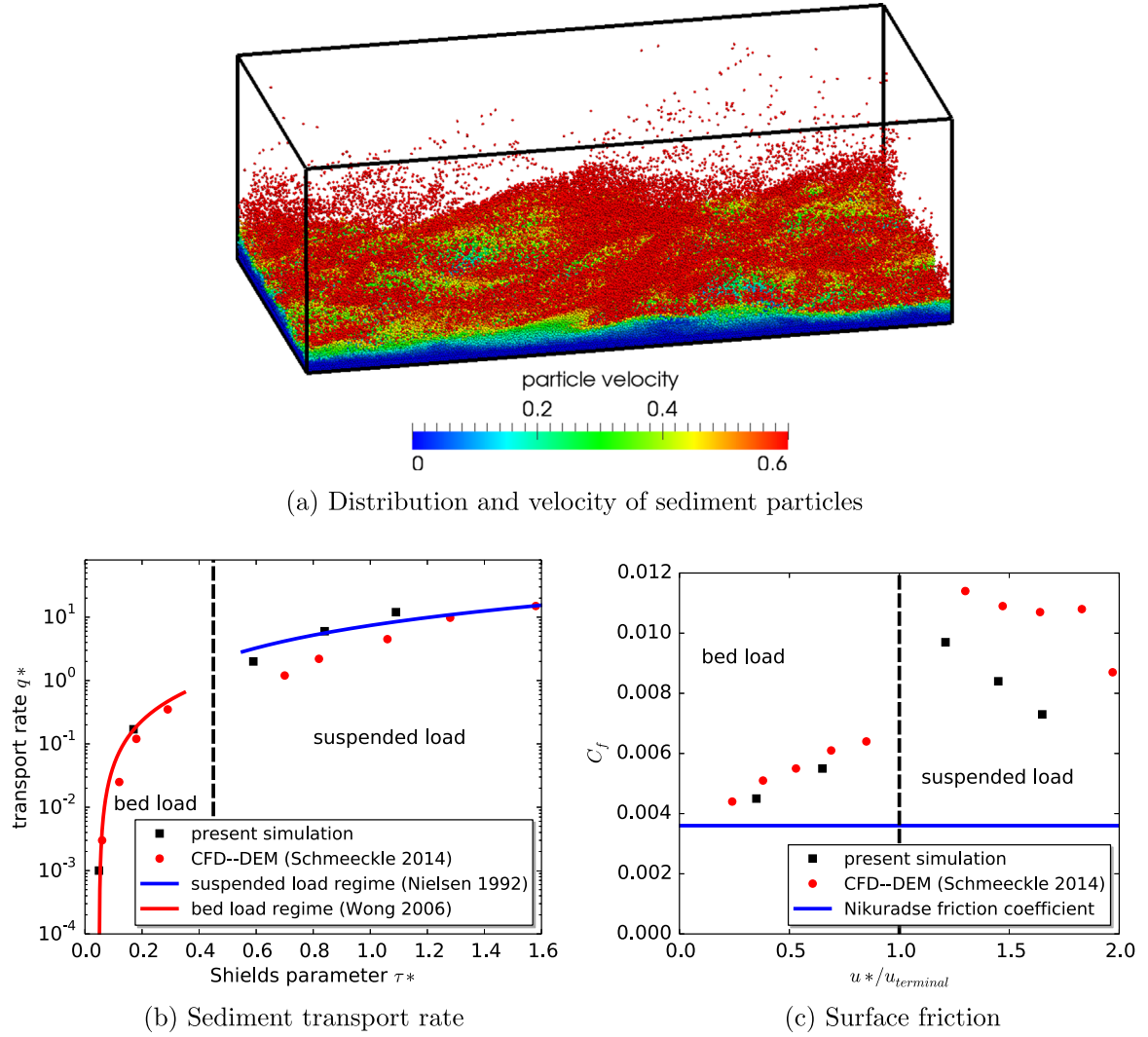

(c) Surface friction

Fig. 11. CFD–DEM simulation of sediment transport in a periodic channel with *SediFoam*, using 330,000 particles. The snapshot of the particles (colored by their velocity magnitudes) and comparison with empirical formulas of bed load transport and suspended load in different regimes (Wong and Parker, 2006; Nielsen, 1992) are displayed, showing favorable agreement. (a) Distribution and velocity of sediment particles, (b) sediment transport rate and (c) surface friction. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)
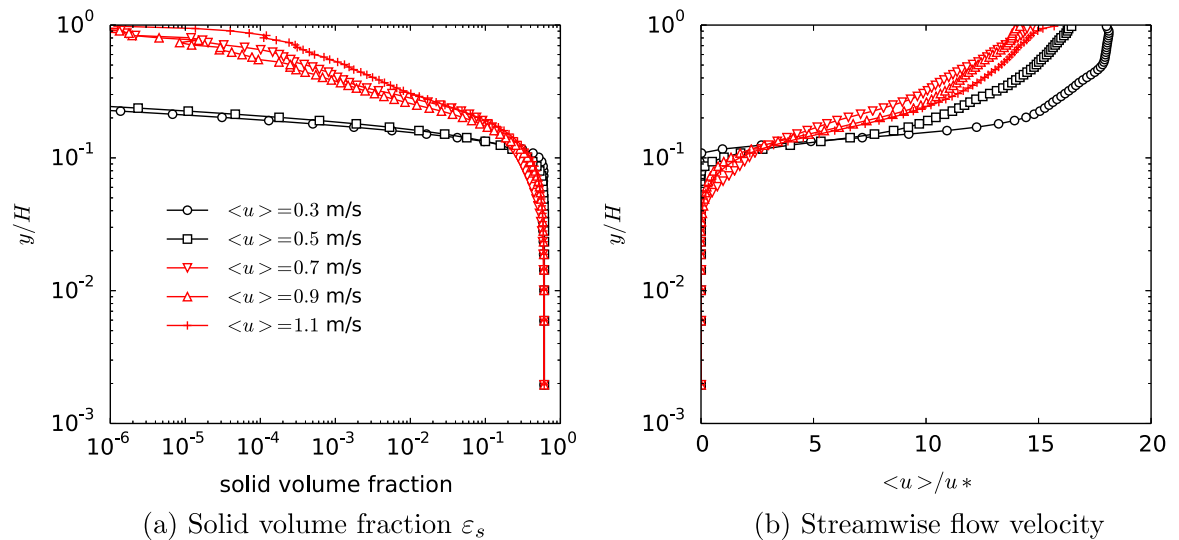

(a) Solid volume fraction ε_s

(b) Streamwise flow velocity

Fig. 12. The solid volume fraction ε_s and streamwise flow velocity of different flow velocities. The blue lines are the simulations in bed load regime; the red lines are the simulations in suspended load regime. (a) Solid volume fraction ε_s and (b) streamwise flow velocity. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

The spatially and temporally averaged profiles of solid volume fraction and flow velocity are shown in Fig. 12. It can be seen from Fig. 12(a) that the ε_s above $y/H = 0.2$ is approximately zero in the bed load regime. This is because the sediment particles are rolling and sliding on the sediment bed in this regime. In the suspended load regime, the particles are suspended in the flow due to turbulent eddies. Therefore, ε_s above $y/H = 0.2$ is much larger. The flow velocity profiles also vary at different regimes of sediment transport. In the bed load regime, the height of the sediment bed is approximately $0.1H$. The sediment particles on the sediment bed are moving slowly in this regime. Hence, the streamwise flow velocity decays rapidly under the sediment bed. In the suspended load regime, the flow velocity at the sediment bed is more diffusive since the particles at the sediment bed are moving more rapidly. Compared with the data obtained in the literature (Schmeeckle, 2014), the overall agreement of solid volume fraction and flow velocity is satisfactory.

5. Scalability

The parallel efficiency is crucial to CFD–DEM solvers for simulations of large and complex problems. In Sections 5.1 and 5.2, the parallel efficiencies in the simulations of fluidized bed and sediment transport are studied separately. This is because the setup, the flow regime, and the behavior of particles are different between the two cases. The CPU time spent on different parts of SediFoam is detailed in Section 5.3.

Both strong and weak scalability are studied in the parallel efficiency tests. The strong scalability is to evaluate the performance of a solver for a constant sized problem being separated by using more and more processors. The amount of the total work is constant but the amount of communication work increases. The parallel efficiency of the strong scalability test is $N_{p0}t_{p0}/N_{pn}t_{pn}$, where N_{p0} and N_{pn} are the number of processors employed in the simulation of the baseline case and the test case, respectively; t_{p0} and t_{pn} are the CPU time. The speed-up is defined as t_{p0}/t_{pn}, which is the relative improvement of the CPU time when solving the problem. The weak scalability evaluates the performance of a solver for the problem with increasing number of processors. In the weak scalability test, the amount of the work in different processors is the same. The parallel efficiency of the weak scalability test is t_{p0}/t_{pn}. The scale-up is defined as $N_{pn}t_{p0}/t_{pn}$, which is the improvement in the scale of the problem that SediFoam can solve.

5.1. Scalability of fluidized bed simulation

The validation case of fluidized bed (Sun and Xiao, 2015a) is applied to test the parallel efficiency of SediFoam. PCM-based averaging, which requires little computational cost, is applied to reduce the influence of averaging algorithm. In the simulation of fluidized bed, various numbers of processors are used from 4 to 256. The computational hours are calculated by running 100 time steps in each case.

To test the strong scalability of the code for large-scale simulations, the numerical test is performed using 5.3 million sediment particles and 1.3 million CFD cells. The domain is 1056 mm × 600 mm × 240 mm in width (x-), height (y-), and transverse thickness (z-) directions. The results obtained in the strong scalability test are shown in Fig. 13(a). It can be seen that the parallel efficiency of SediFoam is close to 100% when employing less than 32 processors and is still as high as 85% when using 128 processors. However, the parallel efficiency decreases to 52% when using 256 processors. In the test using 256 processors, the number of sediment particles in each processor is as small as 20,000. From the results reported in the literature, the parallel efficiency of SediFoam is approximately 15% higher than other solvers when using this number of particles (Gopalakrishnan and Tafti, 2013). It is noted in Fig. 13(a) that the parallel efficiency fluctuates when using 4–32 processors. This may attribute to the fluctuation of the CPU time when solving the linear system. When using different numbers of processors, the decomposition of the domain leads to different linear systems and different converge rates when solving these linear systems. This fluctuation may also be due to the variation of CPU cache or memory. The fluctuation of the CPU time can also be seen in the speed-up curve, but the magnitude is small.

To test the weak scalability of SediFoam, 8.3×10^4 particles are located in each processor with the dimensions of 66 mm × 600 mm × 60 mm in width (x-), height (y-), and transverse thickness (z-) directions. The numbers of processors vary from 4 to 256, and thus the total numbers of particles vary from 0.3 million to 21 million. It can be seen in Fig. 13(b) that good weak scalability is observed. The parallel efficiency is close to 100% when using less than 32 processors and gradually decreases to 61% when using 256 processors. Therefore, the ultimate scale-up of the code by using 256 processors is as large as 156.
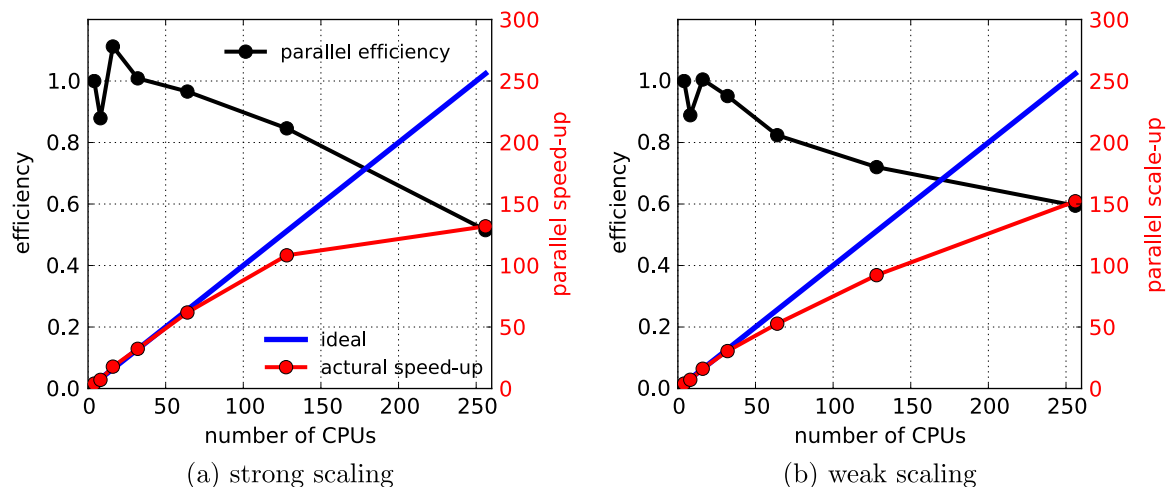

(a) strong scaling   (b) weak scaling

Fig. 13. The parallel efficiency of SediFoam for strong scaling and weak scaling in fluidized bed. The test case employs up to 21 million sediment particles using up to 256 CPU cores. The simulation with the smallest number cores is regarded as base case when computing speed-up. Tests are performed on Virginia Tech's BlueRidge cluster. (http://www.arc.vt.edu/). (a) Strong scaling and (b) weak scaling.
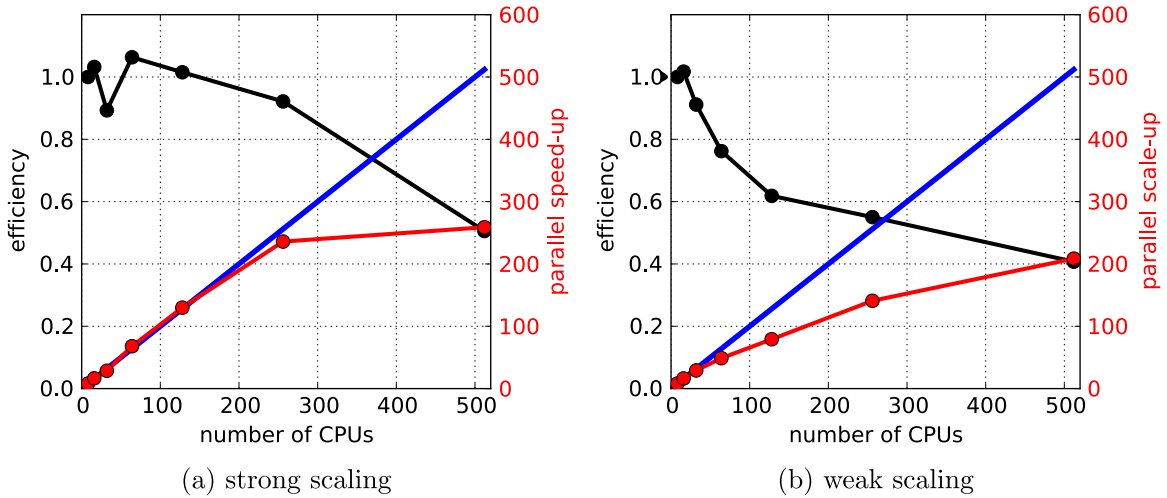
Fig. 14. The parallel efficiency of *SediFoam* for strong scaling and weak scaling in sediment transport. The test case employs up to 40 million sediment particles using up to 512 CPU cores. The simulation with the smallest number cores is regarded as base case when computing speed-up. (a) Strong scaling and (b) weak scaling.

Table 2
Breakdown of computational costs associated with different parts of fluidized bed simulations. For both tests, the CPU times presented here are normalized by the time spent on the CFD part of the case using 256 processors.

	$N_p=4$	$N_p=16$	$N_p=64$	$N_p=256$
Strong scaling				
CFD	33.6(28%)	7.7 (27%)	1.9(23%)	1.0(28%)
DEM	89.0(71%)	20.4(72%)	5.9(74%)	1.6(46%)
Coupling	3.1 (2%)	0.4 (1%)	0.2(3%)	0.9(21%)
Total	125.7	28.5	8.0	3.6
Weak scaling				
CFD	0.4(21%)	0.3(27%)	0.7(26%)	1.0(31%)
DEM	1.6(76%)	1.7(72%)	1.7(70%)	1.9(56%)
Coupling	0.1(3%)	0.0(1%)	0.1(4%)	0.4(13%)
Total	2.1	2.1	2.5	3.3

5.2. Scalability of sediment transport simulation

Simulations of sediment transport cases are performed to test the parallel efficiency of *SediFoam* when the number of CFD mesh is larger than the number of particles. The diffusion-based averaging algorithm is applied in the simulations. The numbers of processors used vary from 8 to 512.

To test the strong scalability of the code for large-scale simulations, we expanded the domain in Case 3 to be 480 mm × 40 mm × 480 mm in width (*x*-), height (*y*-), and transverse thickness (*z*-) directions. The numbers of sediment particles and CFD cells are 10 million and 13 million, respectively. It can be seen in Fig. 14(a) that good scalability and parallel efficiency are obtained in this test. The parallel efficiency is close to 100% when using less than 64 processors and decreases to 52% when using 512 processors.

To test the weak scalability for sediment transport, 7.8×10^4 particles are located in each processor with the dimension of 60 mm × 40 mm × 30 mm in width (*x*-), height (*y*-), and transverse thickness (*z*-) directions. The total numbers of particles vary from 0.6 million to 40 million. The parallel efficiency and the scale-up of *SediFoam* are shown in Fig. 13(b). Compared with the results obtained in the fluidized bed test, the parallel efficiency in sediment transport test is lower. This is because solving the fluid equations is difficult in parallel when the number of CFD cells is very large.

In summary, although the parallel efficiency of *SediFoam* in sediment transport cases is smaller than the fluidized bed cases, the general efficiency is still satisfactory compared to other CFD–DEM solvers (Gopalakrishnan and Tafti, 2013; Amritkar et al.,

2012; Capecelatro and Desjardins, 2013). Moreover, simulations of $O(10^7)$ number of particles are performed, which demonstrate the capability of *SediFoam* in the simulation of relatively large scale problems.

5.3. Parallel performance of different components

The investigation of the computational costs of different components in *SediFoam* is detailed in this section. The study consists of both simulations of fluidized bed and sediment transport, using the results obtained in the tests of Sections 5.1 and 5.2. The time spent on the CFD module, the DEM module, the coupling between the two modules, and the averaging algorithm are discussed.

The results obtained in the fluidized bed simulation are shown in Table 2. It can be seen in the table that the most time-consuming process is solving the collision between the particles, which accounts for 46–76% of the total costs. Solving the fluid equations is the second most time-consuming process, which takes 21–31% of the total computational costs for both strong scaling and weak scaling. It is noted that the coupling process takes less than 2% of the total costs when using less than 64 processors. However, the proportion of this part increases significantly to about 20% when using 256 processors. This is because the time spent on the inter-processor communication increases due to the increase in number of processors used in the simulation. This increase can also be seen in the weak scaling test for both the CFD and DEM modules, which are well-established solvers with good parallel efficiency.

The computational costs of different components in the sediment transport simulation are detailed in Table 3. Since the number of CFD cells is larger than the number of sediment particles, the computational costs of the CFD module are larger than the DEM module. It can be seen that the proportion of CFD part is approximately 21–48% the total computational costs for both strong scaling and weak scaling tests. The DEM part accounts for about 17–37% of the total costs. It is noted that the computational overhead of the averaging also accounts for 20–33% of the total computational overhead. The time spent on the coupling is less than 2% of the total time when using less than 128 processors. However, when using as many as 512 processors the time spend on the communication increases significantly.

6. Conclusion

In this work, the parallelized open-source CFD–DEM solver *SediFoam* is developed with emphasis on the simulation of sediment

Table 3
Breakdown of computational costs associated with different parts of sediment transport simulations. For both tests, the CPU times presented here are normalized by the time spent on the CFD part of the case using 8 processors.

	$N_p=8$	$N_p=32$	$N_p=128$	$N_p=512$
Strong scaling				
CFD	74.4(47%)	20.1(47%)	4.4(45%)	1.0(21%)
DEM	35.8(23%)	10.1(23%)	2.8(29%)	0.8(17%)
Averaging	45.7(29%)	12.9(30%)	2.5(25%)	1.4(29%)
Coupling	1.2 (0%)	0.2 (0%)	0.1(1%)	1.6(33%)
Total	157.0	43.2	8.0	4.7
Weak scaling				
CFD	0.2(29%)	0.3(37%)	0.6(45%)	1.0(48%)
DEM	0.3(37%)	0.3(36%)	0.4(29%)	0.4(20%)
Averaging	0.3(33%)	0.2(27%)	0.3(25%)	0.4(20%)
Coupling	0.0(1%)	0.0(1%)	0.0(1%)	0.2(11%)
Total	0.8	0.9	1.4	2.1

transport. The CFD and DEM modules are based on OpenFOAM and LAMMPS, respectively. The communication between the modules is implemented using parallel algorithm to enable the simulation of large-scale problems.

Numerical validations are performed to test the capability of the present solver. The single particle sedimentation test demonstrates the importance of added mass and lubrication in CFD–DEM simulations. In the numerical simulations with 500 sediment particles, the fluid and particle properties obtained are consistent with the results obtained by interface-resolved method. This indicates that the accuracy of *SediFoam* is desirable. The numerical simulation using $O(10^5)$ particles demonstrates the capability of *SediFoam* in the simulation of various regimes in sediment transport.

Parallel efficiency tests are conducted to investigate the scalability of *SediFoam*. From the test, the scalability of *SediFoam* is satisfactory compared with other existing CFD–DEM solvers. This demonstrates that *SediFoam* is a desirable solver for the simulation of sediment transport of large-scale and complex problems.

Acknowledgment

References

Amritkar, A., Tafti, D., Liu, R., Kufrin, R., Chapman, B., 2012. OpenMP parallelism for fluid and fluid-particulate systems. Parallel Comput. 38 (9), 501–517.

Anderson, T., Jackson, R., 1967. A fluid mechanical description of fluidized beds: equations of motion. Ind. Chem. Eng. Fundam. 6, 527–534.

Ball, R.C., Melrose, J.R., 1997. A simulation technique for many spheres in quasi-static motion under frame-invariant pair drag and Brownian forces. Physica A: Stat. Mech. Appl. 247 (1), 444–472.

Calantoni, J., Holland, K.T., Drake, T.G., 2004. Modelling sheet-flow sediment transport in wave-bottom boundary layers using discrete-element modelling. Philos. Trans. R. Soc. Lond. Ser. A 362, 1987–2002.

Capecelatro, J., Desjardins, O., 2013. An Euler–Lagrange strategy for simulating particle-laden flows. J. Comput. Phys. 238, 1–31.

Cundall, P., Strack, D., 1979. A discrete numerical model for granular assemblies. Géotechnique 29, 47–65.

Deb, S., Tafti, D.K., 2013. A novel two-grid formulation for fluid–particle systems using the discrete element method. Powder Technol. 246, 601–616.

Di Felice, R., 1994. The voidage function for fluid–particle interaction systems. Int. J. Multiph. Flow 20 (1), 153–159.

Drake, T.G., Calantoni, J., 2001. Discrete particle model for sheet flow sediment transport in the nearshore. J. Geophys. Res.: Oceans (1978–2012) 106 (C9), 19859–19868.

Durán, O., Andreotti, B., Claudin, P., 2012. Numerical simulation of turbulent sediment transport, from bed load to saltation. Phys. Fluids (1994–present) 24 (10), 103306.

Ebrahimi, M., 2014. CFD–DEM Modelling of Two-Phase Pneumatic Conveying with Experimental Validation (Ph.D. thesis).

Eppinger, T., Seidler, K., Kraume, M., 2011. DEM–CFD simulations of fixed bed reactors with small tube to particle diameter ratios. Chem. Eng. J. 166 (1), 324–331.

Feng, Y., Yu, A., 2004. Assessment of model formulations in the discrete particle simulation of gas–solid flow. Ind. Eng. Chem. Res. 43 (26), 8378–8390.

Fries, L., Antonyuk, S., Heinrich, S., Palzer, S., 2011. DEM–CFD modeling of a fluidized bed spray granulator. Chem. Eng. Sci. 66 (11), 2340–2355.

Furbish, D., Schmeeckle, M., 2013. A probabilistic derivation of the exponential-like distribution of bed load particle velocities. Water Resour. Res. 49 (3), 1537–1551.

Garg, R., Galvin, J., Li, T., Pannala, S., 2012. Open-source MFIX-DEM software for gas-solids flows: Part I—Verification studies. Powder Technol. 220, 122–137.

Gondret, P., Lance, M., Petit, L., 2002. Bouncing motion of spherical particles in fluids. Phys. Fluids (1994–present) 14 (2), 643–652.

Goniva, C., Kloss, C., Pirker, S., 2009. Towards fast parallel CFD–DEM: an open-source perspective. In: Proceedings of Open Source CFD International Conference, November 12–13, Barcelona, Spain.

Gopalakrishnan, P., Tafti, D., 2013. Development of parallel DEM for the open source code MFIX. Powder Technol. 235, 33–41.

Gupta, 2015. Verification and Validation of a DEM–CFD Model and Multiscale Modelling of Cohesive Fluidization Regimes (Ph.D. thesis).

Hockney, R., Goel, S., Eastwood, J., 1974. Quiet high-resolution computer models of a plasma. J. Comput. Phys. 14 (2), 148–158.

Issa, R.I., 1986. Solution of the implicitly discretised fluid flow equations by operator-splitting. J. Comput. Phys. 62 (1), 40–65.

Jajcevic, D., Siegmann, E., Radeke, C., Khinast, J.G., 2013. Large-scale CFD–DEM simulations of fluidized granular systems. Chem. Eng. Sci. 98, 298–310.

Jasak, H., 1996. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows (Ph.D. thesis). Imperial College London, University of London.

Kafui, K., Thornton, C., Adams, M., 2002. Discrete particle-continuum fluid modelling of gas-solid fluidised beds. Chem. Eng. Sci. 57 (13).

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

Kidanemariam, A.G., Uhlmann, M., 2014a. Direct numerical simulation of pattern formation in subaqueous sediment. J. Fluid Mech. 750, 1–13.

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

Li, T., Garg, R., Galvin, J., Pannala, S., 2012. Open-source MFIX-DEM software for gas-solids flows: Part II—validation studies. Powder Technol. 220, 138–150.

Nielsen, P., 1992. Coastal Bottom Boundary Layers and Sediment Transport. World Scientific Publishing, Singapore, SG.

OpenCFD, 2013. OpenFOAM User Guide. See also ⟨http://www.opencfd.co.uk/openfoam⟩.

Plimpton, J., 1995. Fast parallel algorithms for short-range molecular dynamics. J. Comput. Phys. 117, 1–19, See also ⟨http://lammps.sandia.gov/index.html⟩.

Saffman, P., 1965. The lift on a small sphere in a slow shear flow. J. Fluid Mech. 22 (02), 385–400.

Schmeeckle, M.W., 2014. Numerical simulation of turbulence and sediment transport of medium sand. J. Geophys. Res.: Earth Surf. 119, 1240–1262.

Spogis, Nicholas, 2008. Multiphase modeling using EDEM–CFD coupling for fluent. In: 3rd Latin American CFD Workshop, Applied to the Oil Industry, CFD OIL 2008, August. 18–19, Mirador Rio Hotel, Rio de Janeiro. Brazil.

Sun, J., Battaglia, F., Subramaniam, S., 2007. Hybrid two-fluid DEM simulation of gas-solid fluidized beds. J. Fluids Eng. 129 (11), 1394–1403.

Sun, J., Xiao, H., Gao, D., 2009. Numerical study of segregation using multiscale models. Int. J. Comput. Fluid Dyn. 23, 81–92.

Sun, R., Xiao, H., 2015a. Diffusion-based coarse graining in hybrid continuum-discrete solvers: applications in CFD–DEM. Int. J. Multiph. Flow 72, 233–247.

Sun, R., Xiao, H., 2015b. Diffusion-based coarse graining in hybrid continuum-discrete solvers: theoretical formulation and a priori tests. Int. J. Multiph. Flow 77, 142–157.

Syamlal, M., Rogers, W., O'Brien, T., 1993. MFIX Documentation: Theory Guide. Technical Report. National Energy Technology Laboratory, Department of Energy. See also ⟨http://www.mfix.org⟩.

Tsuji, Y., Kawaguchi, T., Tanaka, T., 1993. Discrete particle simulation of two-dimensional fluidized bed. Powder Technol. 77 (79-87).

van Rijn, L., 1984. Sediment transport, Part I: Bed load transport. J. Hydraul. Eng. 110 (10), 1431–1456.

Wang, R.-Q., Law, A.W.-K., Adams, E.E., 2014. Large-Eddy Simulation (LES) of

settling particle cloud dynamics. Int. J. Multiph. Flow 67, 65–75.

Wen, C., Yu, Y., 1966. Mechanics of fluidization. In: Chemical Engineering Progress Symposium Series, vol. 62, p. 100.

Wong, M., Parker, G., 2006. Reanalysis and correction of bed-load relation of Meyer–Peter and Müller using their own database. J. Hydraul. Eng. 132 (11), 1159–1168.

Wu, C.L., Ayeni, O., Berrouk, A.S., Nandakumar, K., 2014. Parallel algorithms for CFD–DEM modeling of dense particulate flows. Chem. Eng. Sci. 118, 221–244.

Xiao, H., Sun, J., 2011. Algorithms in a robust hybrid CFD–DEM solver for particle-laden flows. Commun. Comput. Phys. 9, 297–323.

Zhou, Z., Kuang, S., Chu, K., Yu, A., 2010. Discrete particle simulation of particle–fluid flow: model formulations and their applicability. J. Fluid Mech. 661, 482–510.

Zhu, H.P., Yu, A.B., 2002. Averaging method of granular materials. Phys. Rev. E 66 (2), 021302.