

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

## Powder Technology

journal homepage: [www.journals.elsevier.com/powder-technology](http://www.journals.elsevier.com/powder-technology)

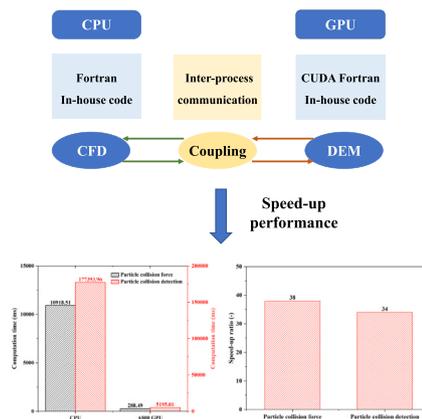
## GPU-accelerated discrete element simulation of granular and gas-solid flows

Jiahui Yu<sup>a</sup>, Shuai Wang<sup>a,\*</sup>, Kun Luo<sup>a,b,\*</sup>, Jianren Fan<sup>a,b</sup><sup>a</sup> State Key Laboratory of Clean Energy Utilization, Zhejiang University, Hangzhou 310027, China<sup>b</sup> Shanghai Institute for Advanced Study of Zhejiang University, Shanghai 200120, China

## HIGHLIGHTS

- GPU-accelerated DEM was coupled with a CFD solver through Message Passing Interface.
- The model is reliable to predict particle dynamics and gas-solid flow patterns.
- GPU-based particle collision parallel algorithm significantly reduces calculation time.

## GRAPHICAL ABSTRACT



## ARTICLE INFO

## Keywords:

CFD-DEM

GPU

Gas-solid flow

Fluidized bed

Numerical simulation

## ABSTRACT

Granular and gas-solid flows are commonly encountered in a range of chemical engineering processes. However, due to the high computational costs, it remains challenging to investigate particle behavior and gas-solid flow hydrodynamics through discrete element simulations. This work developed a graphics processing unit (GPU)-accelerated discrete element method (DEM) that employs an efficient particle collision parallel algorithm and takes full advantage of the parallel structure of GPUs. The DEM code was further coupled with a computational fluid dynamics (CFD) solver through message passing interface (MPI), making it possible to simulate dense gas-solid two-phase flow. The integrated model is verified through three base cases, i.e., a single particle falling and colliding with the wall, two stacked particles compressed between two fixed walls, and a single particle settling in the fluid. The simulation results are in good agreement with the analytic results, indicating the accuracy of the current model. Additionally, this model can accurately predict the particle vertical velocity in a small-scale bubbling fluidized bed and a fully three-dimensional (3D) spout-fluidized bed, confirming its reliability in simulating dense gas-solid flow systems. Furthermore, the GPU-accelerated particle collision parallel algorithm significantly reduces the calculation time and shows great speed-up performance and stability.

\* Corresponding authors at: State Key Laboratory of Clean Energy Utilization, Zhejiang University, Hangzhou 310027, China.

E-mail addresses: [wshuai2014@zju.edu.cn](mailto:wshuai2014@zju.edu.cn) (S. Wang), [zjulk@zju.edu.cn](mailto:zjulk@zju.edu.cn) (K. Luo).<https://doi.org/10.1016/j.powtec.2024.119475>

Received 3 October 2023; Received in revised form 3 January 2024; Accepted 28 January 2024

Available online 1 February 2024

0032-5910/© 2024 Elsevier B.V. All rights reserved.

## 1. Introduction

Granular and gas-solid flows widely exist in a variety of chemical engineering processes including chemical looping combustion, solid fuels gasification, and blast furnace ironmaking [1–4]. These multi-phase systems are characterized by high particle concentration, high-frequency particle collision, and complex inter-phase interaction [5]. For many decades, numerous experimental studies have been carried out for a better understanding of granular and gas-solid flow systems and for the optimization of reactors such as fluidized beds and blast furnaces [6–8]. However, it remains challenging to fully investigate the particle behavior and gas-solid flow hydrodynamics due to the high expense and harsh operating conditions (e.g., high pressure and high temperature).

As an alternative to the experimental method, computational fluid dynamics (CFD) is considered a cost-effective tool to investigate dense gas-solid flow [9]. According to the state-of-the-art review of multi-scale simulations of dense particulate systems [10], the numerical method can be generally classified as Eulerian-Eulerian and Eulerian-Lagrangian methods and the main difference lies in the treatment of solid particles. The former one, also known as the two-fluid method (TFM), regards both gas and solid phases as the continuous phase under the Eulerian framework and simplifies the particle-particle collision. This method can economize computational resources and has been widely applied to investigate the characteristics of dense gas-solid flow within many large-scale reactors [11,12]. However, it cannot capture particle-scale information which limits its further application in simulating complex reactors that aim to process particles [13]. In contrast, the computational fluid dynamics - discrete element method (CFD-DEM) can track each particle's movement under the Lagrangian framework and truly solve particle-particle collision through the soft-sphere contact model which helps obtain abundant particle-scale information [14]. It can not only simulate granular flow by the DEM, but also gas-solid flow by the CFD-DEM coupling framework. Moreover, the CFD-DEM method can be readily extended to integrate with thermochemical sub-models to depict intricate homogeneous and heterogeneous reactions, as well as heat and mass transfer [15]. Thus, it has been widely used to investigate the hydrodynamics and reactive characteristics of various chemical engineering processes. For example, Wang et al. [2,15] simulated biomass gasification in a fluidized bed reactor using the high-fidelity CFD-DEM method with thermochemical sub-models and further studied the effects of key operating parameters on gasification performance. Lin et al. [16] numerically studied the hydrodynamic and thermochemical characteristics in a coal-direct chemical looping combustion (CLC) system using a developed CFD-DEM model featuring a polydisperse drag model, heat and mass transfer, and homogeneous and heterogeneous reactions. Although the CFD-DEM method has been widely developed to study complex hydrodynamics and thermochemical characteristics in quasi-2D or small-scale reactors, it is still difficult to be used to study large-scale reactors within numerous particles due to its unaffordable computational costs.

In recent years, several speed-up strategies for the CFD-DEM method have been developed with the development of computer hardware and numerical algorithms [17–20]. Graphics processing unit (GPU), which highly takes advantage of the parallel structure of GPU, has increasingly attracted researchers' interest. In the simulation of granular flow, the GPU has been utilized for parallel simulations with message passing interface (MPI) for information communication, which can significantly accelerate the computational speed. Based on different granularities of parallel, the GPU-based particle collision calculation method can be divided into particle-parallel approach and collision-pair-parallel approach [17]. For the former, each GPU thread calculates the collision force for one particle which loops over all the neighbor particles and calculates the corresponding particle-neighbor particle collision. However, this may lead to inactive threads as 32 threads in a warp execute the same instructions. For the latter, each thread only calculates the

collision of two particles which can address the warp diverge issue. Furthermore, the calculation times of collisions can be reduced, which performs a better performance. Recently, the GPU combining the central processing unit (CPU) parallel simulation method has been developed for gas-solid flow simulation under the Eulerian-Lagrangian framework. Lu [17] developed a GPU-accelerated DEM code coupling with the open-source CFD software MFIX to simulate granular and multiphase flows. In the fluidized bed simulation, the DEM computation time was reduced from 91% to 17%. A comparison of speed-up performance was conducted, indicating that the collision pair parallel approach achieved approximately 6% - 8% higher speed-up compared to the particle parallel approach. Norouzi et al. [21] developed a new hybrid CPU-GPU solver based on the open-source CFD code OpenFOAM and their in-house DEM code. After model verification, the solver was evaluated in three cases of gas-solid flow systems with different geometries and numbers of particles. For each second simulation of a large system containing 870 k particles, it took about 6 h with two CPU cores while only 30 min for a smaller system with 47 k particles with one CPU core. He et al. [22] developed a novel cross-platform coupling approach integrating a commercial CFD solver, ANSYS Fluent, with a standalone GPU-based DEM solver through network communication. For a fluidized system with 1.3 million particles, the GPU-based DEM provided 3.81 times speedup versus OpenMP on 32 CPU cores.

However, in the traditional CFD-DEM method, the high-frequency particle collision calculation results in low computational efficiency and it is difficult to carry out the massively parallel calculation of particle collision detection processing on CPUs. Thus, the present work has the following novelty: (i) a GPU-accelerated DEM code is developed based on an efficient particle collision parallel algorithm, which takes full advantage of the parallel structure of GPUs; (ii) a GPU-accelerated DEM code is further coupled with a CFD solver through MPI for simulating large-scale granular and two-phase flow reactors and pinned memory is adopted to minimize the data transfer time-consuming. The present work is structured as follows: Section 2 details the governing equations and the GPU implementation details of the particle collision parallel algorithm. Section 3 presents the model verification details under three different cases, i.e., a single particle falling process, two stacked particles compressed between two fixed walls, and the terminal velocity of a particle in the fluid. In Section 4, two sets of actual experimental systems and data are used to further test the accuracy of the model. The speed-up performance and stability assessment of the GPU-accelerated CFD-DEM method are discussed in Section 5. Conclusions are drawn in Section 6.

## 2. Mathematical model

### 2.1. Governing equations for gas phase

In the CFD module, the fluid phase is described using the time-averaged governing equations under the Eulerian framework. The mass and momentum conservation equations are given as [2]:

$$\frac{\partial(\varepsilon_g \rho_g)}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g \mathbf{u}_g) = 0 \quad (1)$$

$$\frac{\partial(\varepsilon_g \rho_g \mathbf{u}_g)}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g \mathbf{u}_g \mathbf{u}_g) = \nabla \cdot \bar{\bar{S}}_g + \rho_g \varepsilon_g \mathbf{g} - \sum_{m=1}^M \mathbf{I}_{gm} \quad (2)$$

where  $\varepsilon_g$  and  $\rho_g$  are the volume fraction and density of the gas phase, respectively.  $\mathbf{u}_g$  is the velocity of the gas phase and  $\mathbf{g}$  is the gravitational acceleration.  $\bar{\bar{S}}_g$  is the stress tensor of the gas phase and  $\mathbf{I}_{gm}$  is the momentum exchange term between the gas phase and the  $m^{th}$  solid phase. The gas phase can be coupled with the solid particles through the gas volume fraction ( $\varepsilon_g$ ) and momentum exchange term ( $\mathbf{I}_{gm}$ ), which are given by:

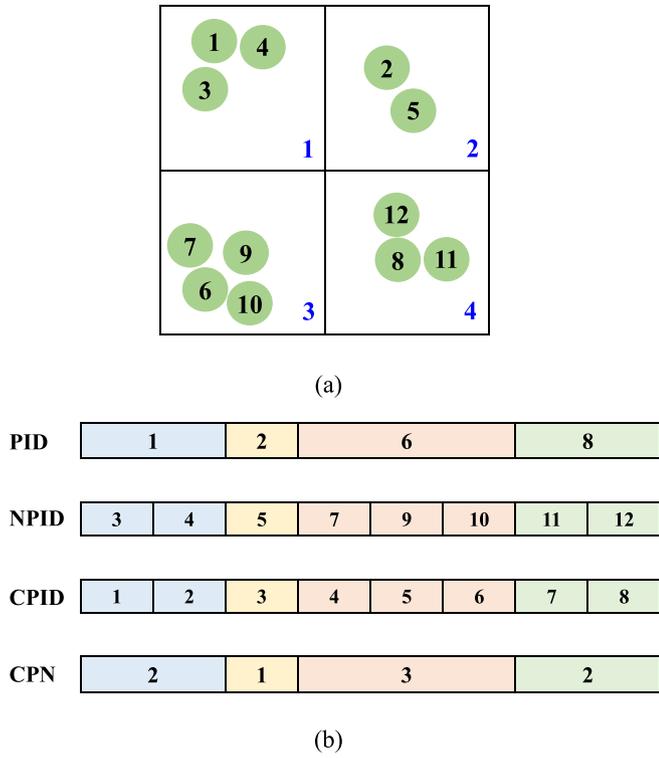


Fig. 1. Schematic of particle collision detection: (a) particles and searching cells; (b) particle neighboring search results.

$$\varepsilon_g = 1 - \frac{1}{V_c} \sum_{i=1}^{N_p} V_{p,i}, \quad \mathbf{I}_{gm} = \frac{1}{V_c} \sum_{i=1}^{N_p} \mathbf{f}_{d,i} \quad (3)$$

where  $V_c$  and  $V_{p,i}$  are the volume of the current computational cell and  $i^{\text{th}}$  particle, respectively.  $N_p$  is the total number of particles in the simulation domain.

## 2.2. Governing equations for solid phase

In the DEM module, the solid particle is tracked individually under the Lagrangian framework. The acceleration, velocity, and position of each particle can be accurately obtained through Newton's second law of motion. The translation and rotation of a particle are described by [23]:

$$m_i \frac{d\mathbf{v}_i}{dt} = m_i \mathbf{g} + \mathbf{f}_{\nabla p} + \mathbf{f}_{d,i} + \mathbf{f}_{c,i} \quad (4)$$

$$I_i \frac{d\boldsymbol{\omega}_i}{dt} = \sum_{j=1, j \neq i}^{N_p} (L\mathbf{n} \times \mathbf{f}_{ct,ij}) \quad (5)$$

where  $m_i$ ,  $\mathbf{v}_i$  and  $\boldsymbol{\omega}_i$  are the mass, translational velocity, and rotational velocity of  $i^{\text{th}}$  particle, respectively.  $L$  is the distance from the contact point to the center of the particle and  $\mathbf{n}$  is the normal unit vector between  $i^{\text{th}}$  and  $j^{\text{th}}$  particles.  $\mathbf{f}_{c,i}$  is contact force exerting on  $i^{\text{th}}$  particle and can be divided into a normal component ( $\mathbf{f}_{cn,ij}$ ) and a tangential component ( $\mathbf{f}_{ct,ij}$ ):

$$\mathbf{f}_{c,i} = \sum_{j=1, j \neq i}^{N_p} (\mathbf{f}_{cn,ij} + \mathbf{f}_{ct,ij}) \quad (6)$$

$$\mathbf{f}_{cn,ij} = - \left( k_{n,ij} \delta_{n,ij} + \eta_{n,ij} \dot{\delta}_{n,ij} \right) \mathbf{n}_{ij} \quad (7)$$

$$\mathbf{f}_{ct,ij} = \begin{cases} - \left( k_{t,ij} \delta_{t,ij} + \eta_{t,ij} \dot{\delta}_{t,ij} \right) \mathbf{t}_{ij} & \text{for } |\mathbf{f}_{ct,ij}| \leq \mu |\mathbf{f}_{cn,ij}| \\ -\mu |\mathbf{f}_{cn,ij}| \mathbf{t}_{ij} & \text{for } |\mathbf{f}_{ct,ij}| > \mu |\mathbf{f}_{cn,ij}| \end{cases} \quad (8)$$

where  $\eta_{n,ij}$  is the normal damping coefficient and can be calculated based on the linear spring dashpot (LSD) model as [24]:

$$e_{n,ij} = \exp \left( - \frac{\eta_{n,ij} t_{n,ij}^{col}}{2m_{eff}} \right) \quad (9)$$

$$t_{n,ij}^{col} = \pi \left( \frac{k_{n,ij}}{m_{eff}} - \frac{\eta_{n,ij}^2}{m_{eff}^2} \right)^{-1/2} \quad (10)$$

$$\eta_{n,ij} = \sqrt{2k_{n,ij}m_{eff}} \frac{|\ln e_{n,ij}|}{\sqrt{\pi^2 + \ln^2 e_{n,ij}}} \quad (11)$$

where  $e_{n,ij}$  is the normal restitution coefficient and  $t_{n,ij}^{col}$  is the collision time.  $m_{eff} = m_i m_j / (m_i + m_j)$  is the effective mass of particle  $i$  and  $j$ .  $k_{n,ij}$  is the normal spring coefficient. The tangential damping coefficient can be calculated similarly. Specifically, the tangential displacement is a history accumulation that begins once the contact initiates. At the initial of the contact, the tangential displacement is calculated as:

$$\delta_t = \mathbf{v}_{ij} \min \left( \frac{|\delta_n|}{\mathbf{v}_{ij} \cdot \boldsymbol{\eta}_{ij}}, \Delta t \right) \quad (12)$$

While at the time of  $t + \Delta t$ , the tangential displacement is calculated as:

$$\delta_t(t + \Delta t) = \delta_t(t) + \mathbf{v}_{ij} \Delta t \quad (13)$$

The calculation of  $\mathbf{f}_{c,i}$  is based on the contact list of particles which is constructed based on the position of particles in each solid time step. The details of the determination of the particle contact list will be discussed in the following part.  $\mathbf{f}_{d,i}$  is the drag force between gas and solid phases exerting on  $i^{\text{th}}$  particle. The correlation proposed by Gidaspow is adopted to calculate the drag force [25]:

$$\mathbf{f}_{d,i} = \frac{1}{V_c} \sum_{i=1}^{N_p} \left( \frac{\beta V_{p,i}}{1 - \varepsilon_g} (\mathbf{u}_g - \mathbf{v}_i) \right) \quad (14)$$

$$\beta = \begin{cases} \frac{150(1 - \varepsilon_g)^2 \mu_g + 1.75(1 - \varepsilon_g) \rho_g |u_g - v_i|}{\varepsilon_g d_p^2} & \varepsilon_g \leq 0.8 \\ \frac{3}{4} \frac{\varepsilon_g (1 - \varepsilon_g) \rho_g |u_g - v_i|}{d_p} C_D e_g^{-2.65} & \varepsilon_g > 0.8 \end{cases} \quad (15)$$

$$C_D = \begin{cases} \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687}) & Re_p < 1000 \\ 0.44 & Re_p \geq 1000 \end{cases} \quad (16)$$

$$Re_p = \frac{\rho_g \varepsilon_g |\mathbf{u}_g - \mathbf{v}_i| d_p}{\mu_g} \quad (17)$$

where  $\beta$  is the momentum exchange coefficient and  $C_D$  is the drag force coefficient for a single particle relying on particle Reynolds number ( $Re_p$ ).

## 2.3. GPU implementation

The calculation speed of the CFD-DEM method is mainly limited by the DEM solver which contains high-frequency particle collisions [21,26]. Specifically, there are a lot of computer graphics algorithms in the calculation process of particle motion, such as particle collision detection processing and particle collision force calculation, which are the most computationally intensive parts of the DEM solver. However, it remains challenging to carry out massively parallel calculation of particle collision detection processing on the CPU. As a special multi-threaded parallel processor, GPU has its unique advantage for efficiently processing large-scale particle information. GPU is composed of plenty of stream multiprocessor (SM) and each SM contains a number of cores

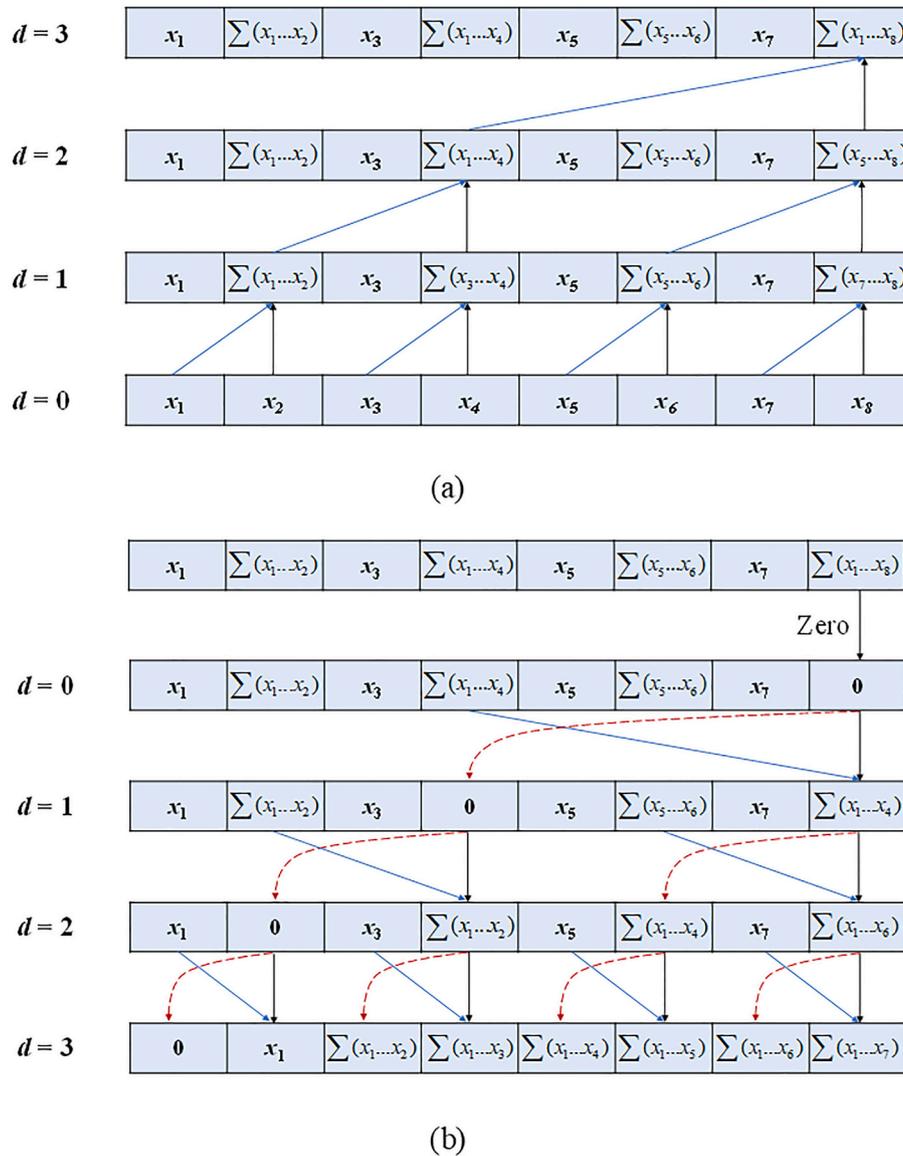


Fig. 2. Illustration of parallel prefix sum algorithm: (a) up-sweep; (b) down-sweep.

that can perform mathematical calculation individually and execute an operation simultaneously. With the introduction of NVIDIA's Compute Unified Device Architecture (CUDA), GPU can execute CUDA C code and CUDA Fortran code and has been widely applied in scientific calculation represented by numerical simulations [17,21]. In addition, OpenACC is an instruction-based programming model and provides a simple way to accelerate without complex programming. Thus, in this work, the particle collision parallel algorithm is implemented on GPU using CUDA Fortran code and OpenACC instruction.

Fig. 1 shows the particle collision detection and particle collision force calculation implemented on GPU with 12 particles. During the simulation, each particle is labeled by an identification number (PID) for the sake of tracking its trajectory and estimating the relative position of two particles. As shown in Fig. 1(a), discrete particles can be mapped into Eulerian grids based on the position of particles through the particle centroid method (PCM). In the CFD-DEM method, the grid size ( $\Delta x$ ) should be 3–5 times the particle diameter ( $d_p$ ). To balance the computational accuracy and efficiency, only particles in spatially adjacent 8 grids are detected to determine whether particle collisions occur. If the particle center distance is less than the sum of the radius of particles, these two particles are considered to collide and should be recorded in

the particle collision list as a collision pair. Furthermore, the same collision pair (i.e., collision pair 1&3 and 3&1) should be recorded only once in consideration of computational efficiency. In this work, the results of particle collision detection are stored in two arrays (one for particle ID in a collision pair and one for collision pair ID) for the particle collision force calculation in the next time-step. As shown in Fig. 1(b), eight collision pairs can be found in this 12-particle system. The collision pair ID (CPID) can be obtained based on the collision pair number (CPN).

Parallel prefix sum is a useful building block for many parallel algorithms including sorting and building data structures and is used to complete the parallel sum of CPN to obtain the array of CPID [27]. To better understand this algorithm, a balanced tree is used, which is often used in parallel computing. Fig. 2 presents the parallel prefix sum algorithm and the algorithm consists of two phases: the up-sweep phase and the down-sweep phase. In the up-sweep phase, the tree traverses from leaf to root and calculates partial sums at the internal node of the tree. After the up-sweep, the last node of the tree is the sum of all nodes in the array. While in the down-sweep phase, the tree traversed back up from the root, using the partial sums to build the scan in place on the array. Note that the total sum of the array is not included in the results,

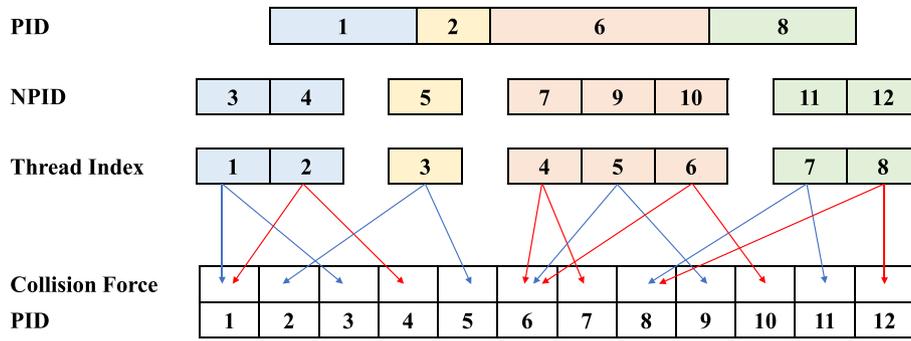
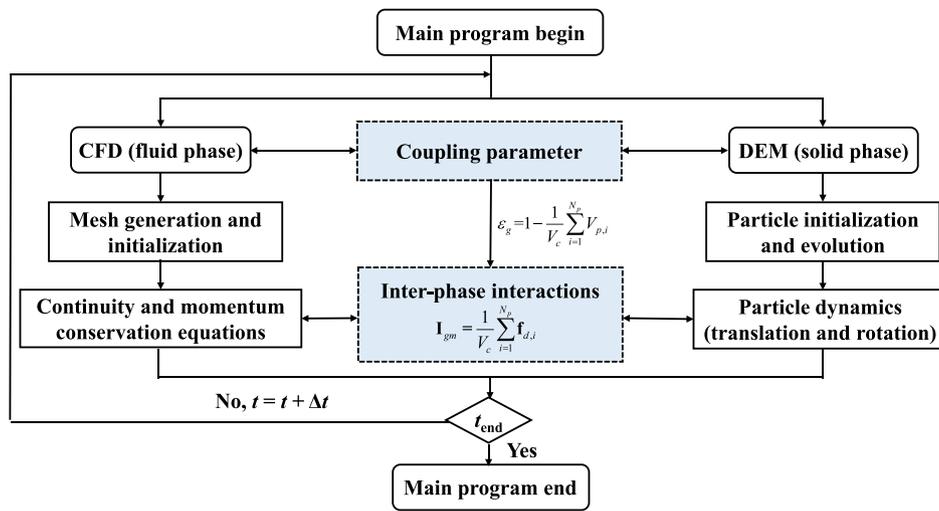
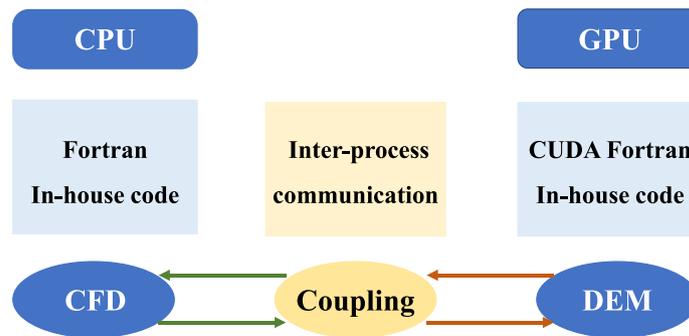


Fig. 3. Schematic of parallel calculation strategy of particle-particle collision force.



(a)



(b)

Fig. 4. Schematic representation of the coupling and GPU acceleration schemes: (a) coupling procedure of the CFD-DEM method; (b) GPU-accelerated CFD-DEM method.

the last element of the array is zeroed at the start of the down-sweep phase. During the down-sweep phase, the zero propagates back to the head of the array. Thus, the list of particle collision pairs can be obtained on GPU accurately and efficiently. Recently, the thrust library developed by NVIDIA provides a standard prefix sum algorithm supported by Tensorcore, which requires the GPU to support Tensorcore. In future work, we will incorporate the standard prefix sum algorithm and compare its performance with the current algorithm.

Fig. 3 illustrates the parallel calculation strategy of the particle collision force. The force can be calculated based on the list of particle

collision pairs and each GPU thread only calculates the collision of two particles based on Eqs. (6–8). According to Newton's third law, the force calculated represents the action and reaction forces acting between two collision particles. For example, the collision force between particle 1 and particle 3 acts on both particles as the action and reaction forces. Thus, the force should be added to particle 1 and subtracted from particle 3. To ensure the accuracy of results, the `acomix_Add` or `acomix_Sub` function of CUDA is used. Once the force calculations are complete, the velocity and position of particles can be updated based on Eq. (4). The key distinction between the particle collision parallel algorithm and the

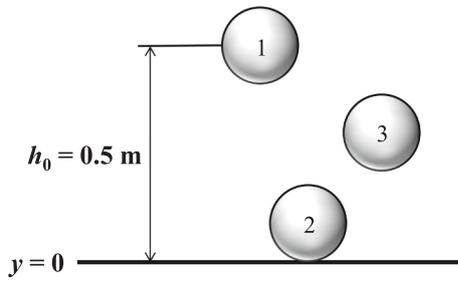


Fig. 5. Schematic of the free falling of a single particle.

collision-pair-parallel algorithm in Lu’s work lies in how the collision pair is calculated. In a previous study [17], the collision pair was constructed from the neighbor list using the set-scan-scatter approach. However, in the present study, the collision pair is generated through a particle collision detection algorithm that relies on the parallel prefix sum algorithm. This algorithm enables efficient identification of collision pairs by analyzing the relative positions of particles in spatially adjacent 8 grids.

2.4. Numerical scheme

Fig. 4 illustrates the coupling scheme of numerical algorithms. The developed GPU-accelerated CFD-DEM model can numerically study granular and gas-solid flows, in which the particle phase is calculated on the GPU using the efficient particle collision parallel algorithm while the fluid phase is solved on the CPU. While in the CFD-DEM simulation, some physical quantities should be transferred between the fluid phase and the solid phase. The data on particle properties are stored in GPU memory while the data on fluid properties are stored in CPU memory. Thus, the information communication is implemented through inter-process communication using pipes which are channels with a writing end and a reading end. Furthermore, pinned memory is adopted to minimize the unavoidable time consumption during data transfer.

The finite volume method (FVM) is used to discretize the governing equations of the gas phase and the first-order upwind scheme is used to discretize other spatial terms. The SIMPLE algorithm is adopted to solve the coupling of the velocity and pressure of the gas phase. Thus, the velocity and pressure of the gas phase can be obtained in each computational node. The Coutant-Friedrich-Lewy (CFL) condition is adopted to

determine the time step of the gas phase ( $\Delta t_{CFD}$ ) from the CFD part [28]:

$$CFL = \Delta t_{CFD} \max \left( \frac{|\mathbf{u}_f|}{\Delta x} \right) < 1 \tag{18}$$

where  $\Delta x$  is the characteristic size of each grid. As for the DEM solver, the time step of the solid phase ( $\Delta t_{DEM}$ ) should be smaller than a critical value to guarantee the numerical stability of multiple collisions. In the work,  $\Delta t_{DEM}$  is specified as 1/50 of the minimum collision time ( $t_{n_{ij}}^{col}$ ) [29].

3. Model verification

Besides the particle collision calculation algorithm, the DEM solver contains a series of complex functions such as particle movement solver, geometry boundary conditions, and the interaction between the CFD solver. The accuracy of the developed code is of great significance and can be verified by comparing the GPU-accelerated results with analytic solutions. To this end, a series of verification studies have been conducted in this section, which consider both granular flow and gas-fluid flow. The aim of these studies is to confirm the accuracy of the GPU-accelerated DEM and CFD-DEM solvers.

3.1. A single particle falling and colliding with the wall

The first verification of the GPU-accelerated DEM solver is to simulate a single particle falling and colliding with the wall. As shown in Fig. 5, a particle with a smooth surface freely falls from a position with an initial height of 0.5 m under gravity and bounces after colliding with

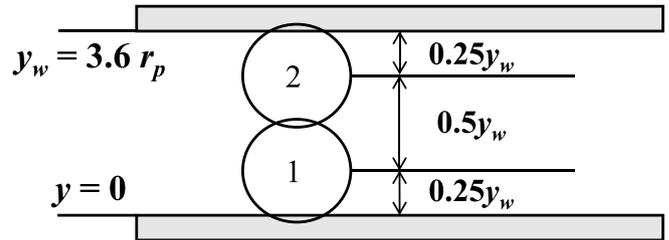
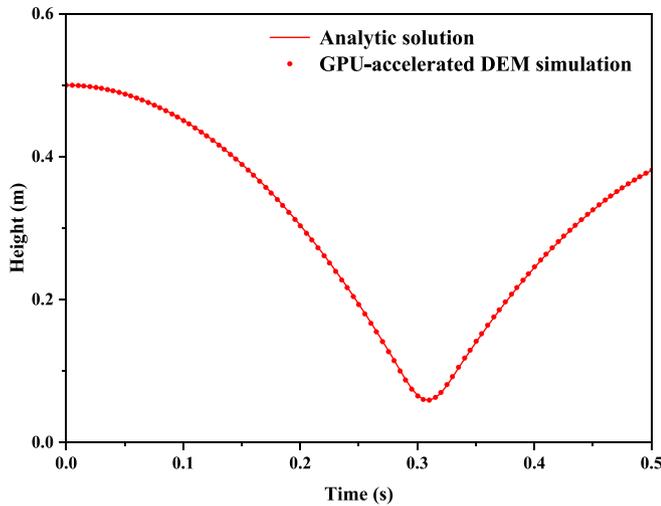
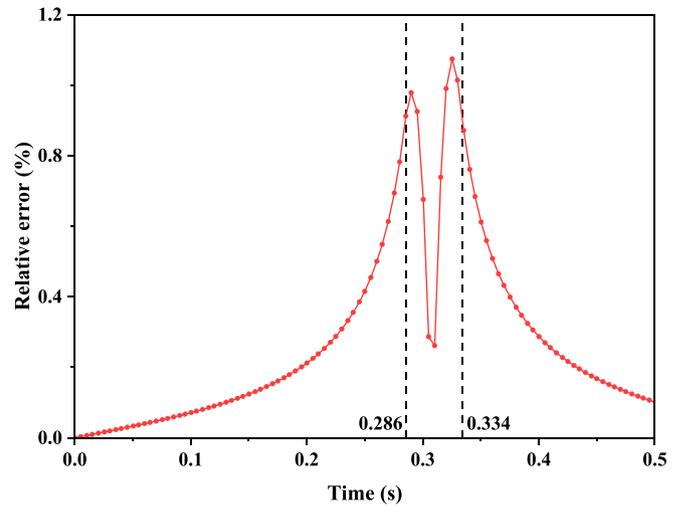


Fig. 7. Schematic of two stacked particles compressed between two boundaries.

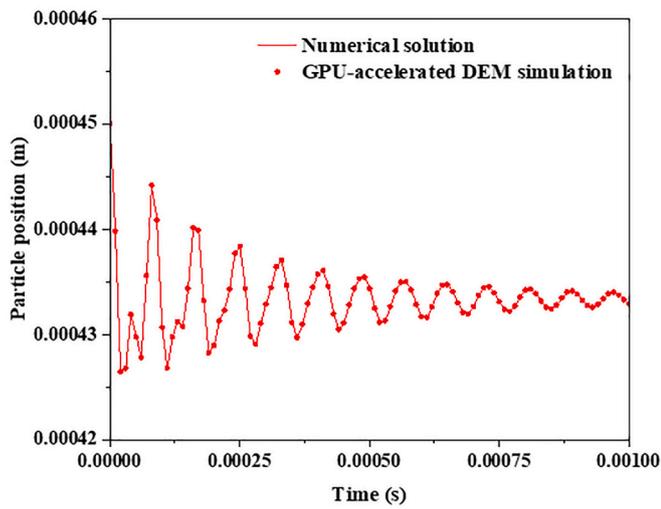


(a)

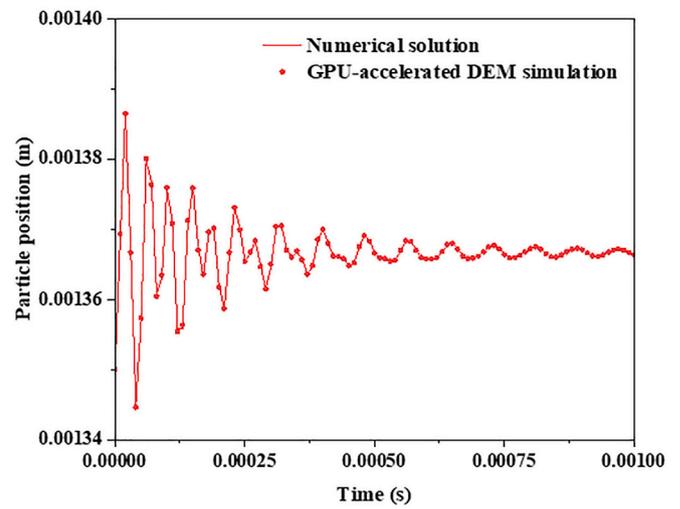


(b)

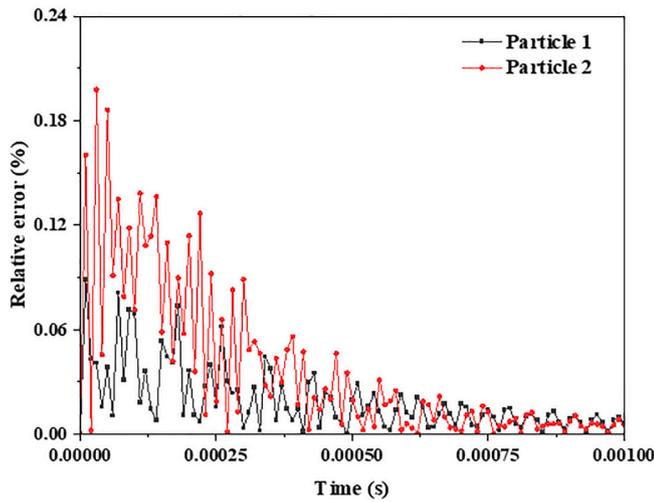
Fig. 6. Comparison of GPU-accelerated DEM simulation results with analytic solutions: (a) time-evolution particle height; (b) relative error.



(a)



(b)



(c)

**Fig. 8.** Comparison of GPU-accelerated DEM simulation results with numerical solutions: (a) time-evolution position of particle 1; (b) time-evolution position of particle 2; (c) relative error.

a fixed wall. The diameter of the particle is 0.2 m. The spring stiffness and restitution coefficient of the particle are  $5 \times 10^4$  N/m and 0.9, respectively. The particle motion can be divided into three stages: free falling, collision, and rebounding. As illustrated in Appendix A in Supporting Information, an analytic expression of particle motion in each stage can be obtained.

Fig. 6(a) compares the particle position between analytic results and simulation results, with the relative error ( $e = \left| \frac{y_a - y_{DEM}}{y_a} \right| \times 100\%$ ) presented in Fig. 6(b). Referring to Appendix A in Supporting Information, the particle colliding the wall at  $t = 0.286$  s and rebounds at  $t = 0.334$  s. Specifically, the particle-wall collision is described using the soft-sphere approach and the linear spring-dashpot model. The simplification of the properties of the particle and the wall may cause the difference between the simulation result and the real result of the duration of the particle-wall contact. The maximum error can be observed in the contact stage which may be caused by a few reasons. In the current simulation, the particle position is updated through a first-order scheme and the error

increases with each time step. Furthermore, there is a hysteresis phenomenon in the judgment of particle motion and particle colliding with the wall before detecting the collision. Thus, the particle is still considered as free falling rather than in contact with the wall. Similarly, the particle is rebounded but still considered in contact with the wall. In general, the absolute relative error is less than 1.2% during all three stages, which is acceptable. Therefore, the GPU-accelerated DEM solve can accurately simulate particle behaviors.

### 3.2. Two stacked particles compressed between two fixed walls

The DEM model is further verified by simulating a system of two stacked particles compressed between two fixed walls as shown in Fig. 7. The diameter of two particles is  $5 \times 10^{-4}$  m while the density is different. The density of the lower and upper particles are  $2 \times 10^4$  kg/m<sup>3</sup> and  $1 \times 10^4$  kg/m<sup>3</sup>, respectively. The lower wall is placed at  $y = 0$  and the upper wall is placed at  $y_w = 3.6r_p = 1.8 \times 10^{-3}$  m. The spring stiffness and restitution coefficient of the particle are  $5 \times 10^4$  N/m and

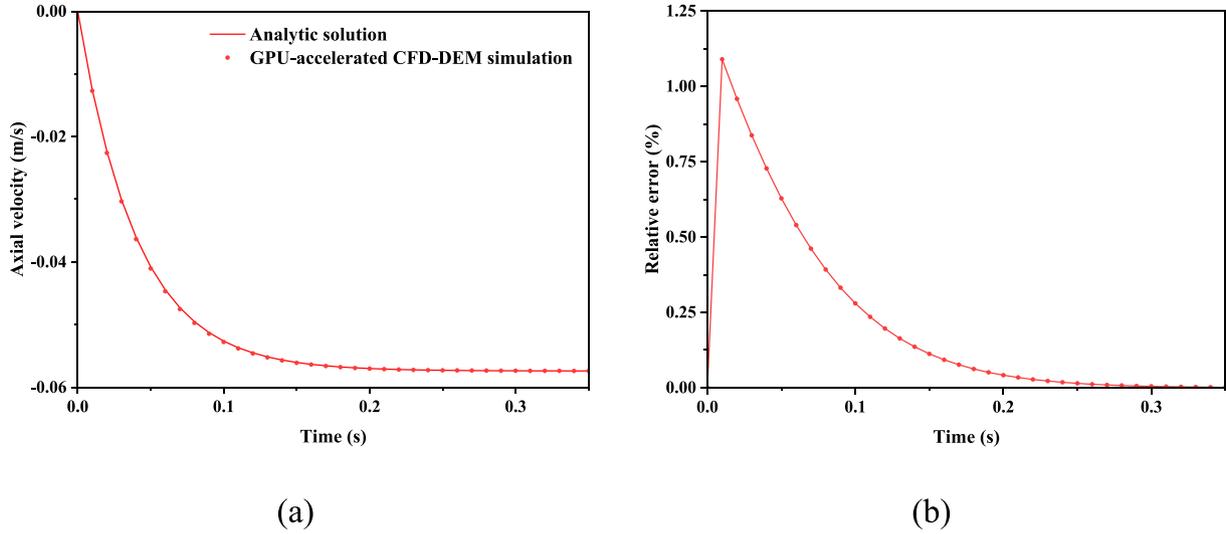


Fig. 9. Comparison of GPU-accelerated DEM simulation results with analytic solutions: (a) particle axial velocity; (b) relative error.

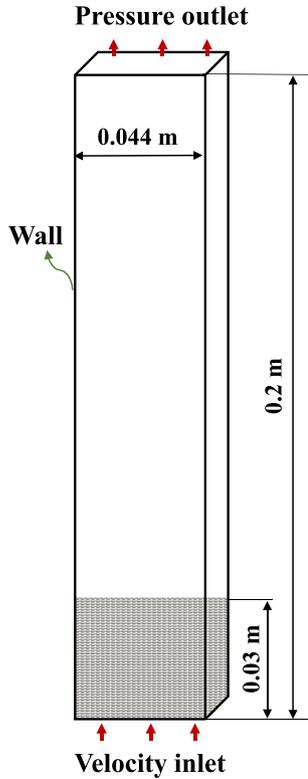


Fig. 10. Geometry configuration of the small-scale bubbling fluidized bed [31].

0.9, respectively. In this system, the particle is always in contact with the wall and another particle, and the motion of particles is dominated by particle-particle and particle-wall collisions. As illustrated in Appendix B in Supporting Information, the total force acting on each particle can be obtained as the basis for comparison. Fig. 8(a) and (b) compare the particle position between numerical solutions and simulation results. As shown in Fig. 8(c), the simulation results agree well with the numerical solutions and the relative error for two particles is less than 0.2%. Therefore, the GPU-accelerated DEM solver can accurately predict particle-particle and particle-wall colliding processes.

Table 1  
Simulation parameters [31].

Parameters	Value	Unit
Bed dimension ( $x, y, z$ )	0.044, 0.01, 0.2	m
Cell number ( $N_x, N_y, N_z$ )	15, 3, 68	–
Packed bed height ( $H_0$ )	0.03	m
<b>Particle phase</b>		
Density ( $\rho_p$ )	1000	kg/m <sup>3</sup>
Diameter ( $d_p$ )	1.2	mm
Number of particle ( $N_p$ )	9240	–
Particle spring stiffness ( $k_n$ )	800	N/m
Particle restitution coefficient ( $e$ )	0.97	–
Particle friction coefficient ( $\mu_p$ )	0.1	–
<b>Gas phase</b>		
Density ( $\rho_g$ )	1.225	kg/m <sup>3</sup>
Viscosity ( $\mu_g$ )	$1.8 \times 10^{-5}$	kg/(m·s)
Superficial velocity ( $U_j$ )	0.6/0.9	m/s

### 3.3. A single particle settling in the fluid

The GPU-accelerated CFD-DEM solver is verified in a gas-solid coupling system, in which the gas and solid phases are coupled via the interphase drag correlation. In this system, the terminal velocity of a particle in the fluid is obtained and the force balance of the particle is given by [30]:

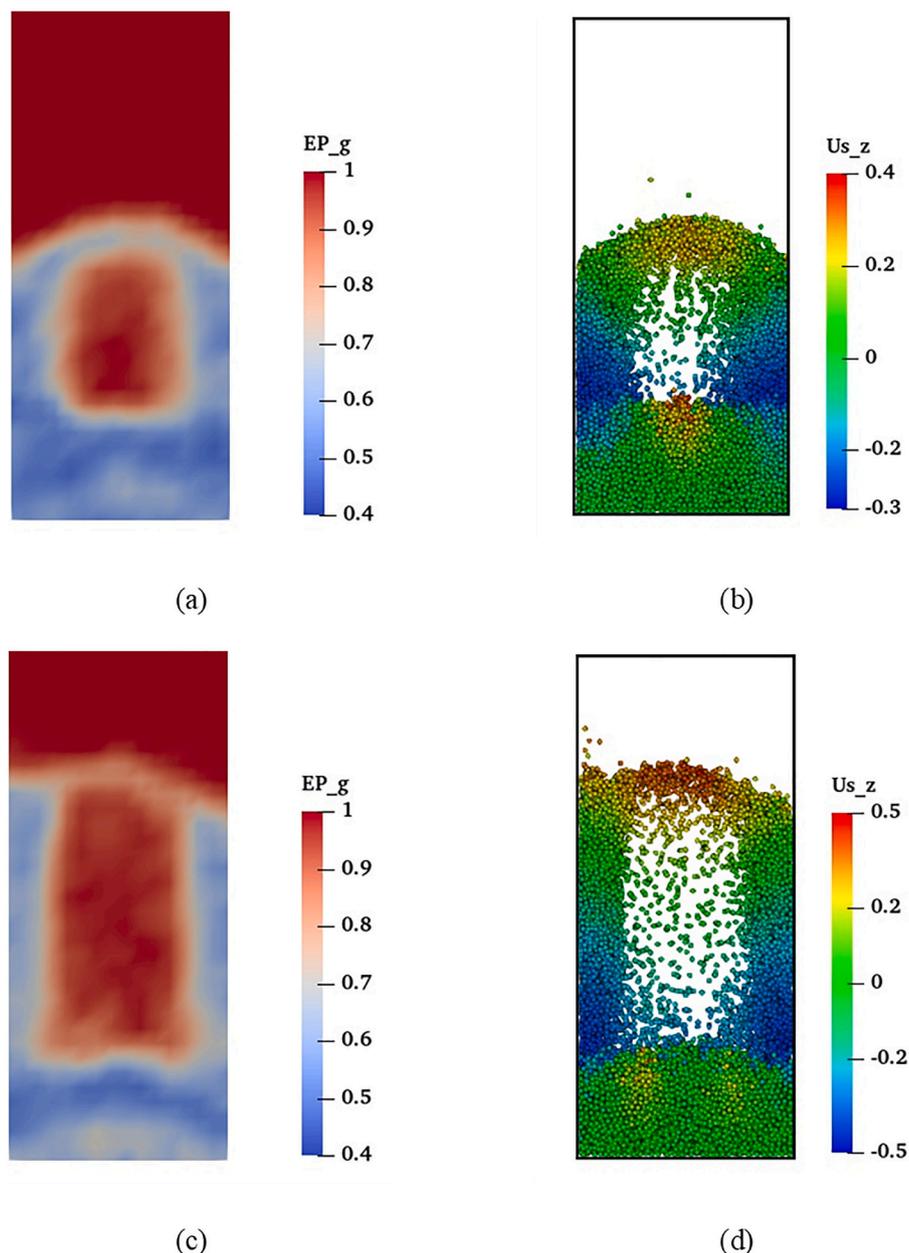
$$\frac{d\mathbf{v}_p}{dt} = -\frac{\mathbf{g}(\rho_p - \rho_g)}{\rho_p} + \frac{3}{4} \frac{\rho_g |\mathbf{u}_g - \mathbf{v}_p| (\mathbf{u}_g - \mathbf{v}_p)}{d_p \rho_p} C_D \quad (20)$$

where  $\rho_p$  and  $\rho_g$  are the density of particle and gas, which are 2000 kg/m<sup>3</sup> and 1.2 kg/m<sup>3</sup>, respectively.  $d_p$  is the diameter of the particle and equals  $1 \times 10^{-4}$  m.  $\mathbf{v}_p$  and  $\mathbf{u}_g$  are the velocities of particle and gas phases, respectively.  $C_D$  is the drag coefficient and can be calculated as:

$$C_D = \frac{24}{Re} (1 + 0.15Re^{0.687}) \quad (21)$$

$$Re = \frac{\rho_g |\mathbf{u}_g - \mathbf{v}_p| d_p}{\mu_g} \quad (22)$$

where  $\mu_g$  ( $=1.8 \times 10^{-5}$  Pa·s) is the viscosity of the gas phase. The fluidized gas is introduced from the bottom of the system with a superficial gas velocity of 0.5 m/s. The particle reaches the terminal velocity when the particle weight is balanced by the drag force. Fig. 9(a)



**Fig. 11.** Snapshots of bubble structure (colored by voidage) and particle flow patterns (colored by particle vertical velocity) in the fluidized bed at  $t = 5$  s:  $U_f = 0.6$  m/s (a, b) and  $U_f = 0.9$  m/s (c, d).

compares the particle axial velocity ( $v_y$ ) between analytic results and simulation results and Fig. 9(b) shows the relative error. The relative error is less than 1.25% and the maximum error can be observed at the initial time. After about  $t = 0.28$  s, the relative error is almost zero, indicating that the terminal velocity of the particle is accurately captured. Therefore, the GPU-accelerated CFD-DEM solver is believed to be accurate in simulating gas-solid flow.

#### 4. Model validation

Although the accuracy of the GPU-accelerated CFD-DEM model has been verified in the above sections, the verification cannot comprehensively reflect the validity of the implemented physical model. Therefore, in this section, two sets of actual experimental systems and data are used to further examine the reasonability of the model.

##### 4.1. Small-scale bubbling fluidized bed

The GPU-accelerated CFD-DEM model is firstly validated by simulating the gas-solid two-phase flow in a small-scale bubbling fluidized bed (BFB) experimentally studied by Müller et al. [31]. As shown in Fig. 10, the dimensions of the investigated system are 0.044 m in width, 0.01 m in depth, and 0.2 m in height, respectively. At the initial time, 9240 particles are packed at the lower part of the bed with a static height of 0.03 m. Referring to the experimental measurement [31], the minimum fluidized velocity is 0.3 m/s, thus two sets of the superficial gas velocities of 0.6 m/s and 0.9 m/s are adopted in the current simulation. Table 1 gives the detailed simulation parameters. The simulation runs for 15 s and the last 10 s of the simulation results are used to make statistics to avoid the startup effects.

Fig. 11 presents the bubble structure and particle flow patterns in the reactor under two different superficial velocities. Large bubbles can be observed to generate in the middle of the bed due to the introduction of

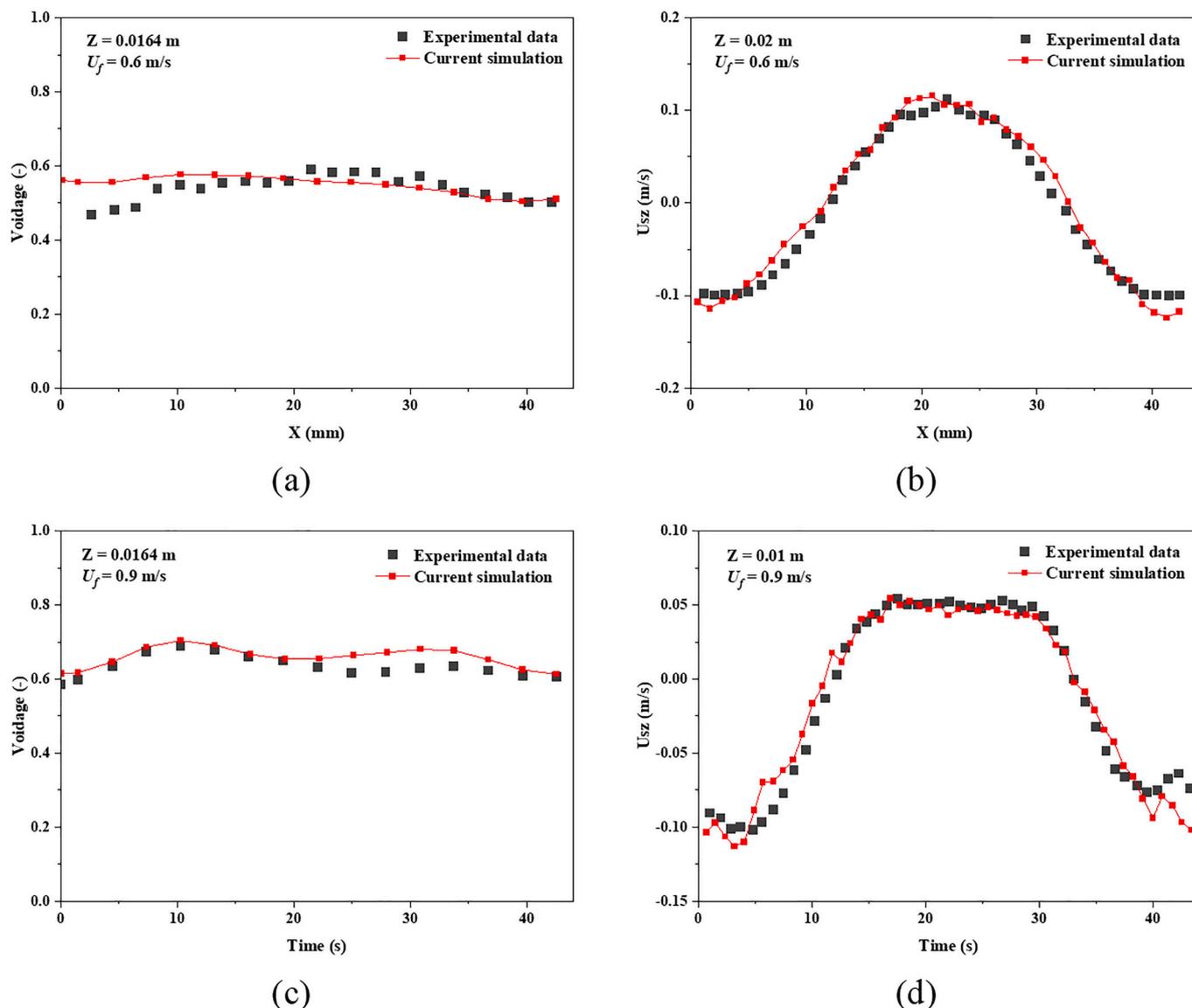


Fig. 12. Comparison of voidage and time-averaged solid vertical velocity between experimental results and current simulation results [31]: (a)  $U_f = 0.6$  m/s,  $Z = 0.0164$  m; (b)  $U_f = 0.6$  m/s,  $Z = 0.2$  m; (c)  $U_f = 0.9$  m/s,  $Z = 0.0164$  m; (d)  $U_f = 0.9$  m/s,  $Z = 0.01$  m.

the fluidized gas and burst near the bed surface. Particles in the central region are in large solid vertical velocity while those appear in the near-wall region are in negative velocity. Fig. 12(a) presents the comparison of the voidage at the height of 0.0164 m and the superficial velocity of 0.6 m/s between experimental results and simulation results. The voidage in most regions can be accurately reproduced, although some discrepancies are observed in the vicinity of the right wall. Fig. 12(b) quantitatively compares the time-averaged solid vertical velocity at the height of 0.02 m, with a good prediction of solid vertical velocity obtained. The large solid vertical velocity can be observed in the central region while the negative solid vertical velocity appears in the near-wall region, which is the back-mixing behavior in the bubbling fluidized bed. In Fig. 12(c) and (d), it can be observed that there is a close correspondence between the voidage at a height of 0.0164 m and the time-averaged solid vertical velocity at a height of 0.01 m when the superficial velocity is 0.9 m/s. Thus, the GPU-accelerated CFD-DEM model can accurately obtain the typical gas-solid characteristics in small-scale BFB.

#### 4.2. Fully 3D spout-fluid bed

The GPU-accelerated CFD-DEM model is further validated with the gas-solid flow in a fully 3D spout-fluid bed experimentally studied by Link et al. [32]. Fig. 13 shows the geometry configuration of the studied system, which has 154 mm in width, 84 mm in depth, and 1000 mm in height. The spouting gas is introduced through the center of the bottom with a width of 22 mm and a depth of 12 mm. The background velocity and spout velocity are 2.5 m/s and 60 m/s, respectively. The detailed operating parameters are listed in Table 2. The simulation runs for 20 s and the last 15 s of the simulation results are used to make statistics to avoid the startup effect.

The spout and annulus regions in 3D spout-fluid beds can be identified by an isosurface by the voidage with a threshold value of 0.7. Fig. 14 shows the comparison of a 3D view of the spout-annulus interface between the experimental results and current simulation results. To eliminate instantaneous fluctuation, a time-averaged result of the current simulation is present in Fig. 14(b) while Fig. 14(a) is an instantaneous experimental result at  $t = 0.08$  s. The current model can well obtain the typical spout-annulus interface in 3D spout-fluid beds.

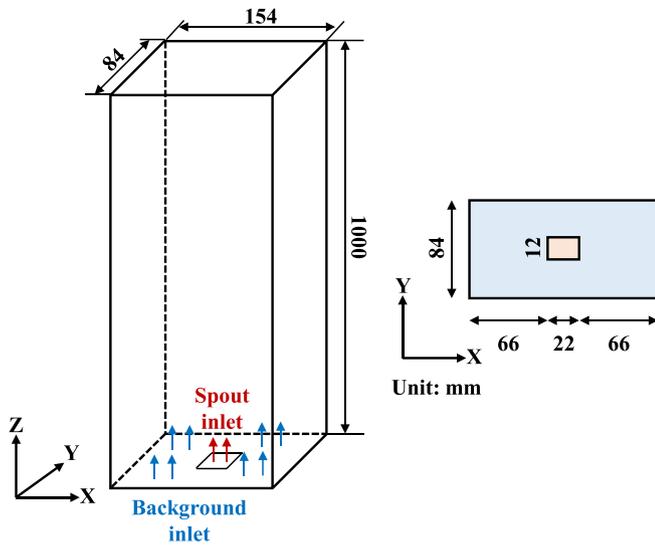


Fig. 13. Geometry configuration of the 3D spout-fluid bed [32].

Table 2  
Simulation parameters [33].

Parameters	Value	Unit
Bed dimension ( $x, y, z$ )	0.154, 0.084, 1.0	m
Cell number ( $N_x, N_y, N_z$ )	21, 14, 100	-
Packed bed height ( $H_0$ )	0.195	m
Particle phase		
Density ( $\rho_p$ )	2526	kg/m <sup>3</sup>
Diameter ( $d_p$ )	4.04	mm
Number of particle ( $N_p$ )	44,800	-
Particle spring stiffness ( $k_n$ )	800	N/m
Particle restitution coefficient ( $e$ )	0.97	-
Particle friction coefficient ( $\mu_p$ )	0.1	-
Gas phase		
Density ( $\rho_g$ )	1.205	kg/m <sup>3</sup>
Viscosity ( $\mu_g$ )	$1.8 \times 10^{-5}$	kg/(m·s)
Spouting velocity ( $u_s$ )	60	m/s
Background velocity ( $u_b$ )	2.5	m/s

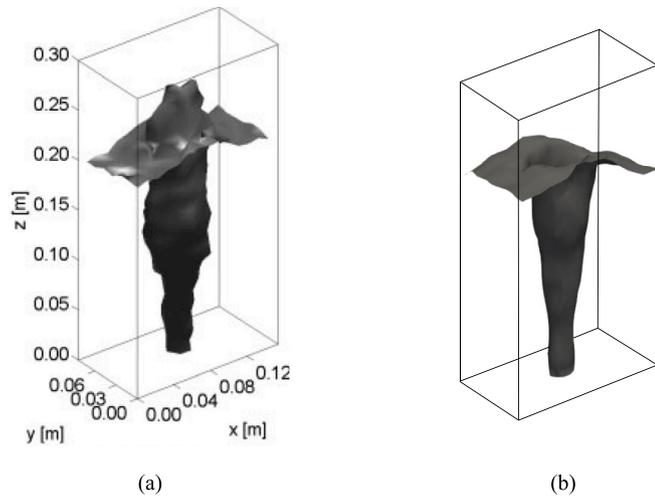


Fig. 14. Comparison of 3D view of the spout-annulus interface: (a) experimental measurement [32]; (b) current simulation.

Fig. 15 shows the time-averaged flow patterns at the central slice ( $y = 0.042$  m). In the central region, the lower voidage and larger solid vertical velocity can be observed due to the introduction of spouting gas. To better illustrate the particle motion characteristics in a 3D spout-fluid bed, particle spatial distribution of the instantaneous hydrodynamics at  $t = 10$  s is presented in Fig. 16. After the spout-fluid bed reaches a stable spouting state, three distinct regions with different solid phase movement behaviors can be observed in the system. In the central region, a small concentration of solid particles can be observed and particles exhibit vigorous upward movement, which is called the spout region. An annulus region surrounds the spout region, where the particles are densely packed and descend slowly. These three regions contribute to the establishment of internal solid particle circulation within the spout-fluid bed.

Fig. 17 presents the quantitative comparison of the time-averaged vertical particle velocity ( $U_{sz}$ ) at two different heights ( $z = 0.15$  m and  $z = 0.25$  m) between current simulation results and the experimental results by Link et al. [32]. A narrow spot channel can be observed in both experimental and simulation results because of the large gas spouting velocity. Similar trends of  $U_{sz}$  can be observed at two different heights, which indicates that the current model can well predict particle vertical velocity in a spout-fluid bed. Thus, the GPU-accelerated CFD-DEM model is reliable for simulating the particle dynamics and flow patterns in 3D spout-fluid beds.

## 5. Performance assessment

### 5.1. Speed-up performance

The speed-up performance of the GPU-accelerated CFD-DEM model is assessed through gas-solid flow in a small-scale BFB. The BFB is 0.23 m in width, 0.075 m in depth, and 1.22 m in height. At the initial time, 95,000 particles are packed in the lower part of the bed. The gas was introduced from the bottom with a superficial gas velocity of 2.1 m/s. To better illustrate the high performance of the particle collision parallel algorithm implemented on GPU, which can be further divided into particle collision detection and particle collision force calculation, the averaged computational time of each time-step is quantitatively statistics. As presented in Fig. 18, the computation time of the GPU-accelerated particle collision force parallel calculation method is greatly shortened, which can obtain 38 times speed-up, showing its good performance and hardware applicability. Furthermore, the GPU-accelerated particle collision detection method also shows great performance which can obtain 34 times speed-up.

### 5.2. Stability assessment

To further illustrate the high performance of the current model, the simulation individually performs in three different ways: CFD-DEM simulation on one CPU core, CFD-DEM parallel simulation on 16 CPU cores, and GPU-accelerated CFD-DEM simulation on one CPU core and one GPU card. The investigated system is 0.5 m in width, 0.5 m in depth, and 1.2 m in height, which consists of 1,000,000 particles. The computational domain is divided into 300,000 grids and the superficial velocity is 0.9 m/s. Each case runs for 1 s and a total of 500 time-steps after 0.5 s is used to perform statistics to avoid the start-up effects. Fig. 19(a) presents the comparison of particle phase computational time consumption under three different calculation conditions. Due to the huge number of particles in the system, the parallel algorithm using 16 CPU cores can obtain nearly 3 times acceleration. Comparable, the GPU-accelerated parallel algorithm can further accelerate the particle calculation process, which is faster than the parallel algorithm using 16 CPU cores. Furthermore, the fluid phase is simply parallelized using the OpenACC multicore parallel function, which is easy to be carried out by adding simple commands without rewriting the calculation codes. As presented in Fig. 19(b), the total CFD calculation time of 500 time-steps

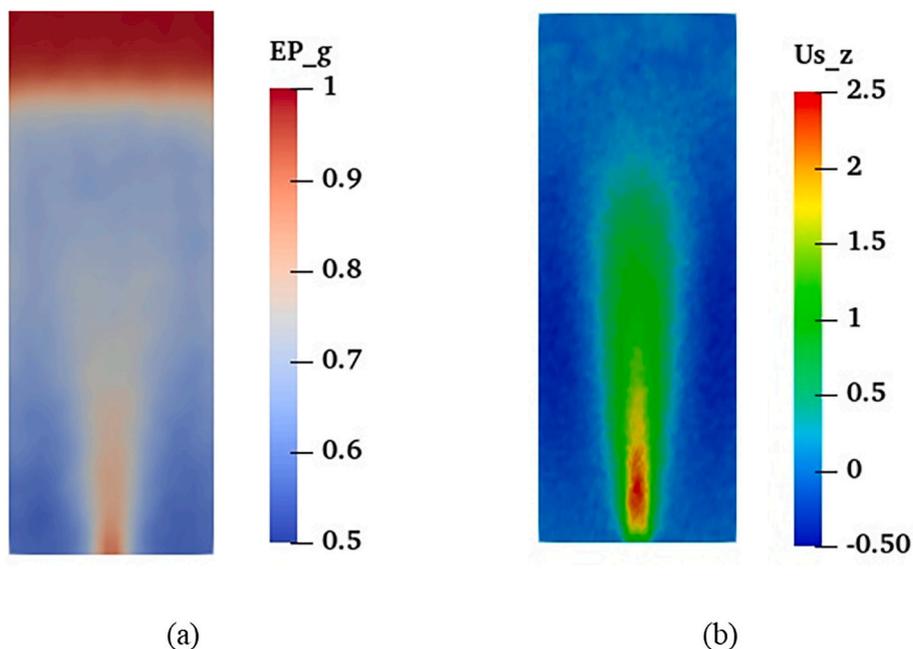


Fig. 15. Snapshots of time-averages flow patterns at the central slice ( $y = 0.042$  m): colored by voidage (a) and solid vertical velocity (b).

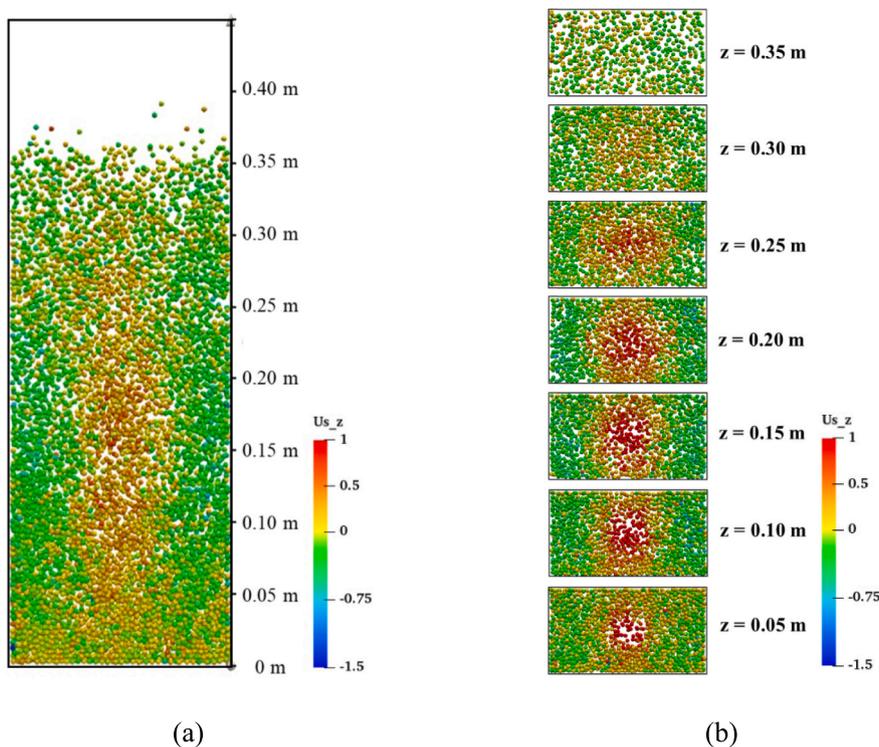
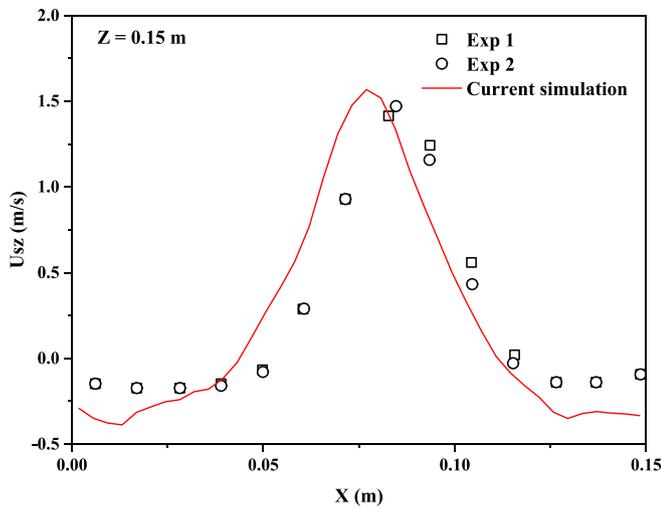


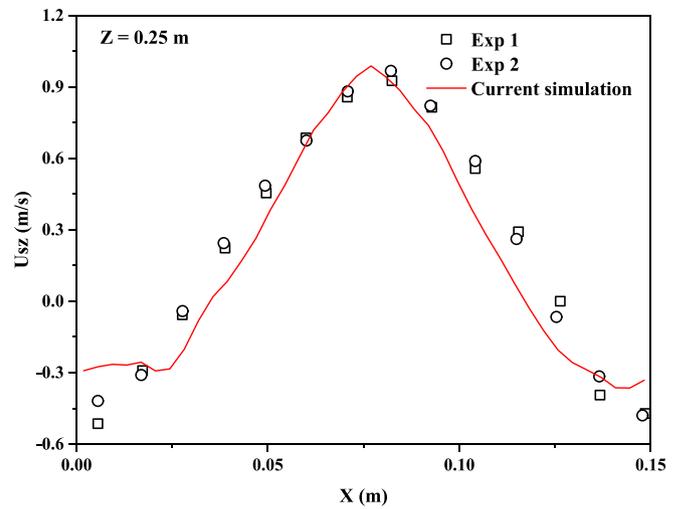
Fig. 16. Snapshots of the spatial distribution of the instantaneous hydrodynamics at  $t = 10$  s (colored by solid vertical velocity): (a) the central slice ( $y = 0.042$  m), (b) slices at different heights.

on CPU can be reduced from 7272 s to 3206 s, obtaining a nearly two times acceleration. Accordingly, the GPU-accelerated CFD-DEM model can obtain a speed-up ratio of approximately 4. Furthermore, the speed-up performance of the GPU-accelerated CFD-DEM model under different numbers of particles is presented in Fig. 19(c). With the increase in particle number, the model demonstrates improved speed-up performance, highlighting the advantageous capability of the current model in efficiently handling a large number of particles. The difference in the

speed-up performance between sections 5.1 and 5.2 is mainly caused by the data transfer between the GPU and CPU. Furthermore, the speed of data transfer is limited by the size of arrays and utilization of the GPU. In our future work, more work needs to be done to break through the limitation of data transfer, such as data memory optimization and asynchronous operation of data transfer and DEM simulation.

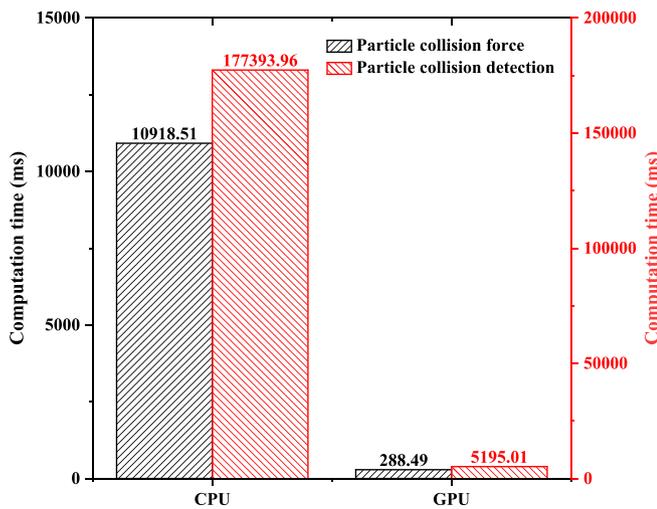


(a)

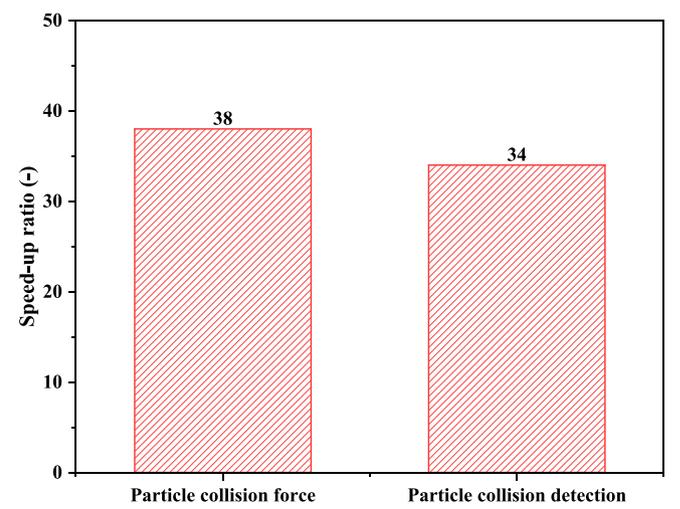


(b)

Fig. 17. Comparisons of the  $U_{sz}$  between current simulation results and experimental results by Link et al. [32] (Exp 1 and Exp 2 are the data obtained by two independent experiments, respectively):  $U_{sz}$  at  $z = 0.15$  m (a) and  $z = 0.25$  m (b).



(a)



(b)

Fig. 18. Performance comparison of GPU-accelerated particle-particle collision force calculation method and the particle collision detection method between the traditional CPU and the A800 GPU: (a) computational time; (b) speed-up ratio.

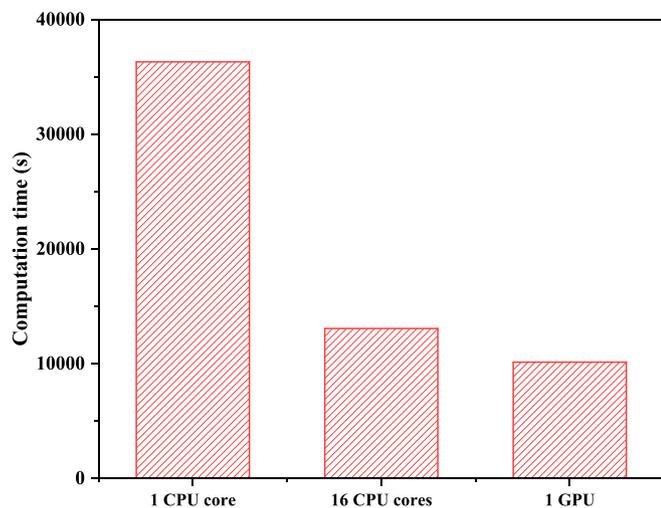
6. Conclusions

In this work, GPU-accelerated DEM/CFD-DEM models were developed based on a high-performance particle collision parallel algorithm for the efficient simulation of granular and gas-solid flows. The accuracy, reliability, and acceleration performance of the developed model are examined under different conditions. Conclusions can be summarized as follows:

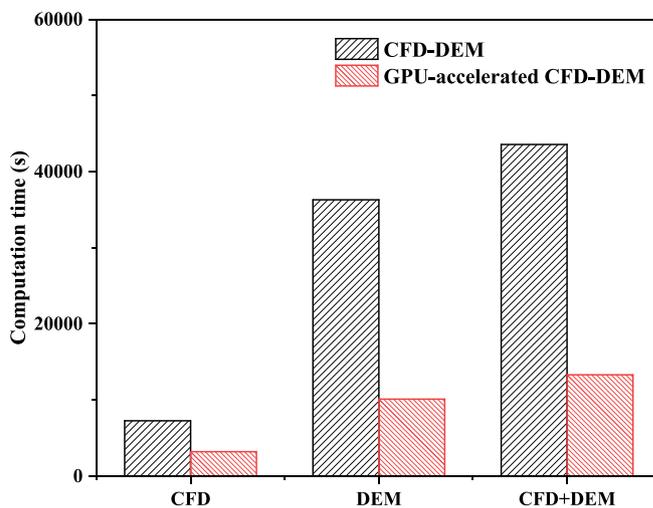
- 1) To fully exploit the parallel architecture of GPUs, we have implemented a GPU-based algorithm for particle collision. This algorithm is integrated into the DEM code, which can be coupled with a CFD solver to simulate gas-solid flows. In this coupled simulation, the particle phase is computed on the GPU, while the fluid phase is solved on the CPU. Particle properties are stored in GPU memory,

whereas fluid properties are stored in CPU memory. The information communication is implemented through inter-process communication and pinned memory is adopted to minimize the data transfer time-consuming.

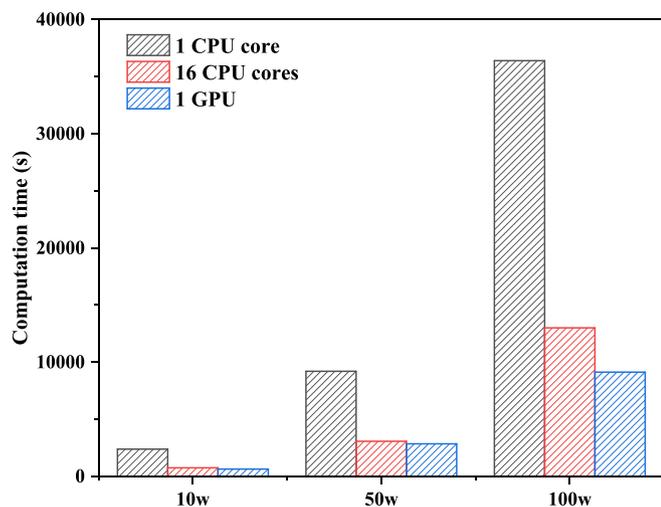
- 2) The present model has demonstrated its reliability in numerically simulating particle-particle and particle-wall collisions, as evidenced by its ability to accurately reproduce the position of two stacked particles compressed between two fixed walls. Additionally, the model has proven effective in predicting the time-averaged vertical particle velocity in small-scale BFB and fully 3D spout-fluidized beds, indicating its capability to accurately capture the particle dynamics and flow patterns in two-phase flow systems.
- 3) The GPU-accelerated particle collision parallel algorithm consists of the particle collision force calculation method and particle collision detection method, which can greatly reduce the calculation time.



(a)



(b)



(c)

Fig. 19. Comparison of particle calculation time consumption: (a) different calculation hardware; (b) different parts; (c) different numbers of particle.

The particle collision force calculation method can obtain 38 times speed-up on A800. The particle collision detection method also obtains 34 times speed-up on A800, which indicates the high-performance of the GPU-accelerated particle collision algorithm.

Thus, this work provides a reliable and high-performance parallel calculation method for numerically studying the gas-solid two-phase flow.

**CRediT authorship contribution statement**

**Jiahui Yu:** Methodology, Software, Validation, Writing – original draft. **Shuai Wang:** Project administration, Writing – review & editing. **Kun Luo:** Project administration, Writing – review & editing. **Jianren Fan:** Project administration.

**Declaration of competing interest**

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

**Data availability**

The data that has been used is confidential.

**Acknowledgments**

We are grateful for the support from the National Natural Science Foundation of China (grant No. 51925603), the Fundamental Research Funds for the Central Universities (2022ZFJH004), and the Shanghai Pujiang Program (23PJ1412000).

## Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.powtec.2024.119475>.

## References

- [1] C. Luo, Z. Peng, E. Doroodchi, B. Moghtaderi, A three-dimensional hot flow model for simulating the alumina encapsulated NI-NIO methane-air CLC system based on the computational fluid dynamics-discrete element method, *Fuel* 224 (2018) 388–400.
- [2] S. Wang, Y. Shen, CFD-DEM study of biomass gasification in a fluidized bed reactor: effects of key operating parameters, *Renew. Energy* 159 (2020) 1146–1164.
- [3] S. Wang, Y. Shen, CFD-DEM modelling of raceway dynamics and coke combustion in an ironmaking blast furnace, *Fuel* 302 (2021) 121167.
- [4] Y. Xu, J. Musser, T. Li, B. Gopalan, R. Panday, J. Tucker, G. Breault, M.A. Clarke, W.A. Rogers, Numerical simulation and experimental study of the gas-solid flow behavior inside a full-loop circulating fluidized bed: evaluation of different drag models, *Ind. Eng. Chem. Res.* 57 (2018) 740–750.
- [5] W. Zhong, A. Yu, G. Zhou, J. Xie, H. Zhang, CFD simulation of dense particulate reaction system: approaches, recent advances and applications, *Chem. Eng. Sci.* 140 (2016) 16–43.
- [6] C. Loha, P.K. Chatterjee, H. Chattopadhyay, Performance of fluidized bed steam gasification of biomass-modeling and experiment, *Energy Convers. Manag.* 52 (2011) 1583–1588.
- [7] J. Ma, D. Mei, C. Wang, X. Tian, Z. Liu, H. Zhao, Sulfur fate during in-situ gasification chemical looping combustion (IG-CLC) of coal, *Chem. Eng. J.* 406 (2021) 126773.
- [8] R.B. Bates, A.F. Ghoniem, W.S. Jablonski, D.L. Carpenter, C. Altantzis, A. Garg, J. L. Barton, R. Chen, R.P. Field, Steam-air blown bubbling fluidized bed biomass gasification (BFBBG): multi-scale models and experimental validation, *AIChE J.* 63 (2017) 1543–1565.
- [9] F. Alobaid, N. Almohammed, M.M. Farid, J. May, P. Rößger, A. Richter, B. Eppler, Progress in CFD simulations of fluidized beds for chemical and energy process engineering, *Prog. Energy Combust. Sci.* 91 (2022) 100930.
- [10] S. Wang, C. Hu, K. Luo, J. Yu, J. Fan, Multi-scale numerical simulation of fluidized beds: model applicability assessment, *Particuology* 80 (2023) 11–41.
- [11] J. Lin, K. Luo, L. Sun, S. Wang, C. Hu, J. Fan, Numerical investigation of nickel-copper oxygen carriers in chemical-looping combustion process with zero emission of CO and H<sub>2</sub>, *Energy Fuel* 33 (2019) 12096–12105.
- [12] Z. Li, H. Xu, W. Yang, S. Wu, Numerical study on the effective utilization of high sulfur petroleum coke for syngas production via chemical looping gasification, *Energy* 235 (2021) 121395.
- [13] J. Wang, Continuum theory for dense gas-solid flow: a state-of-the-art review, *Chem. Eng. Sci.* 215 (2020) 115428.
- [14] M.A. van der Hoef, M. van Sint Annaland, N.G. Deen, J. Kuipers, Numerical simulation of dense gas-solid fluidized beds: a multiscale modeling strategy, *Annu. Rev. Fluid Mech.* 40 (2008) 47–70.
- [15] S. Wang, K. Luo, J. Fan, CFD-DEM coupled with thermochemical sub-models for biomass gasification: validation and sensitivity analysis, *Chem. Eng. Sci.* 217 (2020) 115550.
- [16] J. Lin, K. Luo, S. Wang, L. Sun, J. Fan, Particle-scale study of coal-direct chemical looping combustion (CLC), *Energy* 250 (2022) 123859.
- [17] L. Lu, GPU accelerated MFIX-DEM simulations of granular and multiphase flows, *Particuology* 62 (2022) 14–24.
- [18] L. Lu, S. Benyahia, T. Li, An efficient and reliable predictive method for fluidized bed simulation, *AIChE J.* 63 (2017) 5320–5334.
- [19] L. Zhu, X. Chen, B. Ouyang, W. Yan, H. Lei, Z. Chen, Z. Luo, Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors, *Ind. Eng. Chem. Res.* 61 (2022) 9901–9949.
- [20] S. Pirker, T. Lichtenecker, Efficient time-extrapolation of single-and multiphase simulations by transport based recurrence CFD (rCFD), *Chem. Eng. Sci.* 188 (2018) 65–83.
- [21] H.R. Norouzi, R. Zarghami, N. Mostoufi, New hybrid CPU-GPU solver for CFD-DEM simulation of fluidized beds, *Powder Technol.* 316 (2017) 233–244.
- [22] Y. He, F. Muller, A. Hassanpour, A.E. Bayly, A CPU-GPU cross-platform coupled CFD-DEM approach for complex particle-fluid flows, *Chem. Eng. Sci.* 223 (2020) 115712.
- [23] H.P. Zhu, Z.Y. Zhou, R.Y. Yang, A.B. Yu, Discrete particle simulation of particulate systems: theoretical developments, *Chem. Eng. Sci.* 62 (2007) 3378–3396.
- [24] P.A. Cundall, O.D. Strack, A discrete numerical model for granular assemblies, *Géotechnique* 29 (1979) 47–65.
- [25] D. Gidaspow, Multiphase flow and fluidization: continuum and kinetic theory description, *J. Non-Newton. Fluid Mech.* 55 (1994) 207–208.
- [26] S. Wang, K. Luo, S. Yang, C. Hu, J. Fan, Parallel LES-DEM simulation of dense flows in fluidized beds, *Appl. Therm. Eng.* 111 (2017) 1523–1535, <https://doi.org/10.1016/j.applthermaleng.2016.07.161>.
- [27] M. Safari, M. Huisman, Formal verification of parallel prefix sum and stream compaction algorithms in CUDA, *Theor. Comput. Sci.* 912 (2022) 81–98.
- [28] J.H. Ferziger, M. Perić, R.L. Street, *Computational Methods for Fluid Dynamics*, Springer, 2002.
- [29] Y. Tsuji, T. Kawaguchi, T. Tanaka, Discrete particle simulation of two-dimensional fluidized bed, *Powder Technol.* 77 (1993) 79–87.
- [30] R. Garg, J. Galvin, T. Li, S. Pannala, Open-source MFIX-DEM software for gas-solids flows: part I—verification studies, *Powder Technol.* 220 (2012) 122–137, <https://doi.org/10.1016/j.powtec.2011.09.019>.
- [31] C.R. Müller, S.A. Scott, D.J. Holland, B.C. Clarke, A.J. Sederman, J.S. Dennis, L. F. Gladden, Validation of a discrete element model using magnetic resonance measurements, *Particuology* 7 (2009) 297–306, <https://doi.org/10.1016/j.partic.2009.04.002>.
- [32] J.M. Link, N.G. Deen, J. Kuipers, X. Fan, A. Ingram, D.J. Parker, J. Wood, J. Seville, PEPT and discrete particle simulation study of spout-fluid bed regimes, *AIChE J.* 54 (2008) 1189–1202.
- [33] S. Yang, K. Luo, M. Fang, J. Fan, K. Cen, Influences of operating parameters on the hydrodynamics of a 3-D spout-fluid bed based on DEM modeling approach, *Chem. Eng. J.* 247 (2014) 161–173, <https://doi.org/10.1016/j.cej.2014.03.003>.