Physics of Fluids

RESEARCH ARTICLE | NOVEMBER 27 2023

Calculation of particle volume fraction in computational fluid dynamics-discrete element method simulation of particulate flows with coarse particles

Yan Zhang (张岩) ; Wan-Long Ren (任万龙) ; Peng Li (李鹏) ; Xu-Hui Zhang (张旭辉) ; Xiao-Bing Lu (鲁晓兵)

Physics of Fluids 35, 113330 (2023)
https://doi.org/10.1063/5.0176521

Physics of Fluids ARTICLE pubs.aip.org/aip/pof

Calculation of particle volume fraction in computational fluid dynamics-discrete element method simulation of particulate flows with coarse particles

Cite as: Phys. Fluids **35**, 113330 (2023); doi: 10.1063/5.0176521
Submitted: 14 September 2023 · Accepted: 1 November 2023 ·
Published Online: 27 November 2023

Yan Zhang (张岩), Wan-Long Ren (任万龙), Peng Li (李鹏), Xu-Hui Zhang (张旭辉),[a] and Xiao-Bing Lu (鲁晓兵)

AFFILIATIONS

Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China

[a]Author to whom correspondence should be addressed: zhangxuhui@imech.ac.cn

ABSTRACT

Computational fluid dynamics-discrete element method is frequently used for modeling particulate flows due to its high efficiency and satisfactory accuracy. The particle volume fraction is a crucial parameter that significantly affects the computation accuracy. It may be extremely large when the particulate flows contain coarse particles because it is determined by the ratio of particle volume to cell volume. In this paper, the performance of different methods, such as the divided particle volume method (DPVM), the big particle method, and the diffusion-based method, for computing the particle volume fraction is thoroughly reviewed, implemented, and investigated. It turns out that the DPVM must not be used when the particle size is larger than cell size due to significant fluctuation of the particle volume fraction field. The big particle method is optimized for simulation accuracy and code implementation. The optimized big particle method is similar to the diffusion-based method by diffusing the particle effects to the surrounding cells. It demonstrates greater consistency with experimental observations compared to the diffusion-based method, primarily attributed to its incorporation of polydisperse effects.

I. INTRODUCTION

Particulate flows are widely encountered in natural phenomena and engineering applications, including hydraulic/pneumatic conveying, hydraulic fracturing, debris flows, and fluidized bed.[1–3] From a mechanical perspective, particulate flows exhibit complex dynamics, with shocks between particles, contacts, and shedding of small vortices occurring at the particle scale, and the formation of clusters and plugs at larger scales. The coexistence of different mechanisms at different scales makes particulate flows complex.[4]

Numerical simulation plays an important role in investigating the particulate flows. Numerical models for particulate flows can be continuum-based or discrete-based according to different treatments of the particles. Using a two-phase mixture as an illustrative example, for continuum-based treatment, the model is called the Euler–Euler method, and for discrete-based treatment, the model is called the Euler–Lagrangian method. Gas/fluid and particles are treated as interpenetrating continua in the Euler–Euler method with the solid volume fraction assumed to be uniform at the scale of the computational cell. The two-fluid model (TFM)[5] is a typical Euler–Euler method, which has been widely used to investigate the large-scale particulate flows, such as the centrifugal separation,[6] submarine mudflow,[7] and hydraulic conveying,[8] due to its computational efficiency and convenience. However, the simulation accuracy depends on the closure relations for the particle phase, such as the particle pressure and particle viscosity, which is still an open topic.[9] The problems associated with the Euler–Euler method can be overcome by the Euler–Lagrangian method, where all the particles are tracked individually by taking into account particle–particle and particle–wall collisions. It does not need the complex closure relations for the particle phase in the Euler–Lagrangian method and thus can be applied to a wide range of flow systems.[4,10–12] The Euler–Lagrangian method can be classified into two subcategories: (a) unresolved computational fluid dynamics-discrete element method (CFD-DEM) and (b) particle-resolved simulations (PRS).[13,14] However, enormous computational costs associated with compulsory requirements of fine grids pose serious limitations on the employment of PRS in simulation where the number of particles

reaches tens of thousands.[15] If the particle properties, particularly particle trajectories, are important and the number of particles are larger than tens of thousands, CFD-DEM is one possible choice.

The CFD-DEM combines the advantages of TFM and PRS methods. In CFD-DEM, the continuum phase and discrete phase are coupled via the particle volume fraction and the two-phase interaction forces. The two-phase interaction forces, such as the drag force,[16,17] the virtual mass force, and the lift force,[18] are also related to the particle volume fraction. Hence, the calculation of the particle volume fraction is of vital importance for ensuring computational accuracy in simulation results when using CFD-DEM coupling techniques. The most commonly used method for calculating the particle volume fraction is the particle centroid method (PCM), where the particle volume fraction is obtained by dividing the total volume of particles within a CFD cell by the cell volume. In some applications where the particle size is large, such as the hydraulic conveying in deep sea mining, the ratio of the particle diameter to the pipe diameter is greater than 0.1, which results in particle size comparable to or larger than CFD cell size.[19] In this case, the particle volume fraction in the governing equations of continuum phase is often overestimated and discontinuous, which makes the numerical simulation unstable. Deen *et al.*[20] found that the PCM can guarantee the accuracy of the porosity (or fluid volume fraction) in unresolved CFD-DEM when the ratio of cell size to particle size is larger than 5.0. Hence, to deal with coarse particles, many submodels for calculating volume fraction of coarse particles were proposed, such as the divided particle volume method (DPVM),[21,22] the two-grid method,[23] the statistical kernel method (SKM)[24,25] or the diffusion-based method,[26,27] the virtual mass distribution function method (VMDF),[28] and the big particle method.[29] The DPVM estimates the portions of particle's volume in cells to overcome the large fluctuations in particle/fluid volume fraction. The SKM, diffusion-based method, and VMDF redistribute the particle volume fraction from the host cells to the neighboring cells. However, the calculation results of these methods are influenced by the cell size and the particle size. Therefore, making a good selection of cell size is vital for simulations of particulate flows with coarse particles.

So far, there has been little research systematically comparing the performance of these methods or submodels. Volk, Ghia, and Stoltz[30] used uniform and non-uniform meshes with different sizes to simulate a stationary particle bed and found that a better solution was obtained when the cell size was three times the particle size. Zhang *et al.*[31] compared the two-grid method, the SKM, and the diffusion-based method. They developed an averaging strategy and proposed that the diffusion length[26] in the diffusion-based method should be around three times of the particle size. The purpose of this work is to review the methods of calculating coarse particle volume fraction and to compare their performance in different cell and particle sizes. Some existing methods are also optimized for improved simulation accuracy and code implementation. The scope of applicability of each method is also discussed in detail. In addition, according to Wang, Teng, and Liu,[32] when the particle size is comparable to the CFD cell size, special treatment is required not only for the particle volume fraction but also for the reconstruction of the particle background field. The particle background field is the continuum phase field at the particle centroid, which is related to the field value at the particle's surrounding grid cells. When the coarse particles are viewed as points to the fluid, point particle correlations must be employed to calculate the interphase interactions. For instance, the drag force correlations are utilized to determine the momentum exchange. These interphase interactions, modeled with correlations, will bring two challenges. One is that the correlations require the flow fields to be sampled from a region that is even larger than the particle scale.[33] The other is that the interphase interactions must be projected back onto the fluid appropriately as source or sink terms to the fluid governing equations at the particle's surrounding grid cells. Hence, we also develop a simple method for estimating the background fluid field.

The paper is organized as follows. First, different methods for calculating the volume fraction of coarse particles are reviewed. Some of the methods are optimized, and code implementation for all the methods is also introduced. Then, the governing equations of CFD-DEM are presented briefly. This will be followed by describing the numerical method adopted for solving the governing equations. Finally, three numerical tests are conducted. The application range of different methods is thoroughly studied, and some suggestions are given based on the simulation results.

II. VOLUME FRACTION CALCULATION AND CODE IMPLEMENTATION

The models for calculating the volume fraction of coarse particles mentioned above are reviewed with their origins, principles, advantages, and shortcomings in this section. Then, all the models are implemented in the open-source code CFDEM,[29] which couples the open-source codes OpenFOAM[34] and LIGGGHTS.[29] The DPVM and the original big particle method have been implemented in CFDEM. Other methods are implemented in this paper.

A. Divided particle volume method

The DPVM was proposed by Wu, Berrouk, and Nandakumar,[21] Wu *et al.*[22] for addressing the drawbacks of the PCM, where the exact volume fraction of a particle i in a fluid cell j is accurately determined

$$\omega_{ij} = \frac{V_{ip,j}}{V_{ip}}, \tag{1}$$

where $V_{ip,j}$ is the exact portion of volume of particle i in the fluid cell j, V_{ip} is the volume of particle i, ω_{ij} is used to calculate the particle volume fraction of the concerned fluid cells, and its value is in the range of 0 and 1. When the particle i is entirely located in the fluid cell j, $\omega_{ij} = 1.0$; when it is completely outside that cell, $\omega_{ij} = 0$. Then, the particle volume fraction in the fluid cell j is calculated as

$$\alpha_{p,j} = \sum_{i=1}^{k} \frac{\omega_{ij} V_{ip}}{V_{c,j}}, \tag{2}$$

where k is the number of particles in fluid cell j, which is defined as the particle centroid in cell j. The DPVM helps to smooth exchange fields without artificially enlarging the spatial influence of a particle.[35] However, when the CFD cell is of arbitrary shape, the DPVM requires additional calculations of the particle volume divided by the cell faces, which can be tedious to implement in the code. Even when the CFD cells are cubic, there are also several scenarios where particles intersect with cells that need to be considered. In addition, if the particles are too large, such as being two times the cell size, the DPVM will result in unphysically large particle volume fractions.

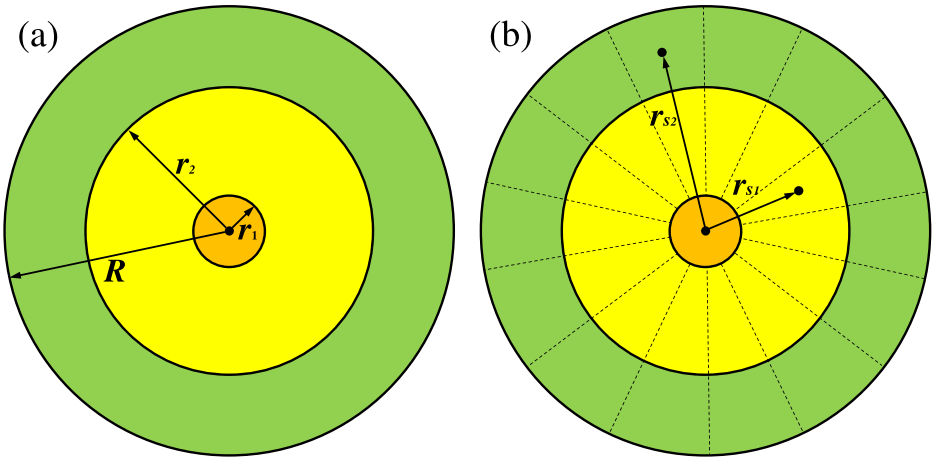

FIG. 1. Schematic representation of the DPVM. The particle is divided into 29 non-overlapping regions of equal volume.

For easily code implementation, a series of distributed marker points are used to resolve a particle. The particle i is divided into 29 non-overlapping regions of equal volume. The centroids of these volumes are used as marker points to reproduce each volume. First, the particle is divided into three parts as shown in Fig. 1(a), where inner part is a sphere with the center coinciding with the particle center. Then, the outer two spherical layers are divided into 28 regions with same volume to the inner sphere [Fig. 1(b)]. Radius of this subsphere is calculated as $r_1 = R^3\sqrt{1/29}$ and $r_2 = R^3\sqrt{15/29}$, where R is the particle radius. Position of the centroid point in radial direction of each volume in the first spherical layer is as follows:

$$r_{s1} = \frac{\int r dr}{\int dr} = \frac{\int_{r_1}^{r_2} r^3 dr}{\int_{r_1}^{r_2} r^2 dr} = \frac{3}{4}\frac{r_2^4 - r_1^4}{r_2^3 - r_1^3}. \tag{3}$$

Similarly, the centroid point for the second spherical layer is

$$r_{s2} = \frac{3}{4}\frac{R^4 - r_2^4}{R^3 - r_2^3}. \tag{4}$$

The positions of all the marker points can be derived from spherical coordinates. The number of marker points in the fluid cell j is determined using the cell topology (*findSingleCell* function in CFDEM). Then, $\omega_{ij} = n_{ij}/29$, where n_{ij} is the number of marker points of particle i in fluid cell j.

B. Two-grid method

The two-grid method was proposed by Deb and Tafti[23] for coarse particles. The particle volume fraction and the CFD simulation are taken in different meshes in the two-grid method, and the two meshes are constructed independently. The particle mesh is chosen according to the particle size to ensure that the ratio of the cell size to the particle size is larger than 3.0. The CFD mesh is set based on the flow resolution requirements. Figure 2(a) gives the schematic of the two-grid method, where the green part is the particle mesh and the black part is the CFD mesh. The drawback of the two-grid method is that its implementation is not straightforward because using two meshes will increase the difficulty of the implementation and necessitate interpolations between the two meshes.[26]

In this paper, we simplify the two-grid method to avoid the data exchange between the CFD mesh and particle mesh. The surrounding fluid cell labels of a particle are obtained according to the cell topology first. The searching region is a sphere with the center coinciding with the particle center and radius three times that of the particle. The linked-list search algorithm[36] is employed to acquire the surrounding cells. In the OpenFOAM software, the inter-cell connections are recorded during the creation of the domain mesh, thereby establishing knowledge of the adjacent cells for each individual cell. For a particle whose centroid is in cell j, we use the *cellcell*() function in OpenFOAM to obtain the adjacent cells of cell j. Next, the distances between the center of an adjacent cell (referred to as cell k) and the centroid of the particle are computed and subsequently compared to the radius of

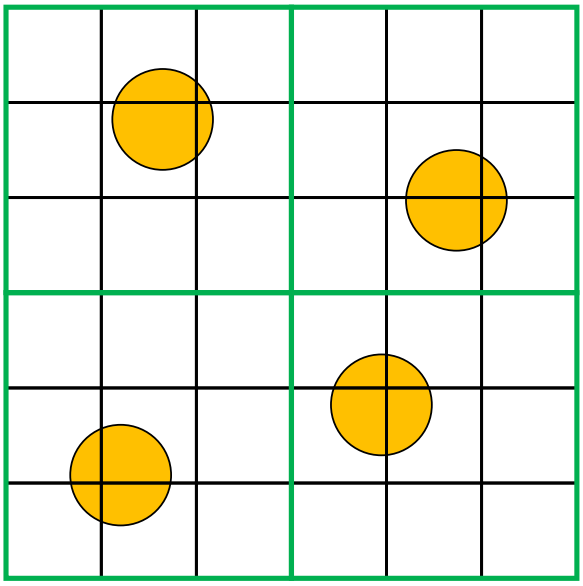

FIG. 2. Schematic representation of the two-grid method.

the search region. If the distance is less than the radius of the search region, the adjacent cell k is recorded as a surrounding cell. Afterward, we examine the adjacent cells of cell k and employ a similar comparison procedure. Finally, the total surrounding cells are searched. It saves much computation effort. The particle volume fraction in all the surrounding fluid cells is same, which is calculated by dividing the particle volume by the sum of the surrounding fluid cell volumes. The particle volume fraction in the fluid cell j is

$$\alpha_{p,j} = \sum_{i=1}^{n} \frac{V_{ip}}{n_i}, \tag{5}$$

where n is the number of the surrounding fluid cells of the particle i. The simplified two-grid method can also be applied in unstructured meshes.

C. Statistical kernel method

The SKM diffuses the coarse particles effects to the surrounding CFD cells. For example, the Gaussian kernel is expressed as[24]

$$h(\mathbf{x}-\mathbf{x}_i) = \frac{1}{(b^2\pi)^{1.5}} \exp\left[-\frac{(\mathbf{x}-\mathbf{x}_i)^{\mathrm{T}}(\mathbf{x}-\mathbf{x}_i)}{b^2}\right], \tag{6}$$

where $\mathbf{x}$ is the position vector, $\mathbf{x}_i$ is particle center coordinates, and b is the bandwidth of the Gaussian kernel. The SKM is easy for code implementation and fast in computation. However, special processing is required at the boundary of the computational domain. Sun and Xiao[26] proposed the diffusion-based method to overcome the drawbacks of the SKM, which is equivalent to the Gaussian kernel method but without special processing at the boundary. A transient, homogeneous diffusion equation is solved in the computational domain that is initiated with the particle volume fraction given by the PCM. This method can be applied to different mesh sizes and is easily parallelizable.

The diffusion equation is written as

$$\frac{\partial \alpha_p}{\partial \tau} = \nabla^2(D\alpha_p), \tag{7}$$

where τ is pseudo-time (time multiplied with a unit diffusion coefficient), which should be distinguished from the physical time t in the CFD-DEM formulation. D is the diffusion coefficient. First, the particle volume in each cell is summed up and the particle volume fraction field α_p is obtained by dividing corresponding cell volumes. Second, the diffusion time T is determined based on desired Gaussian kernel bandwidth, which is in turn specified based on the particle diameter, $b = nd$. To be consistent with the SKM, the diffusion time should be chosen as $b = \sqrt{4T}$ or equivalently $T = b^2/4$. It is recommended that $b = 6d$.[26] Third, the diffusion equation (7) is solved to pseudo-time T to obtain the smoothed particle volume fraction field (as shown in Fig. 3). A Neumann boundary condition with no-flux (zero gradient) has to be specified at all physical boundaries (inlet, outlet, and walls) to ensure the conservation of the particle volume fraction. In this paper, the value of the diffusion coefficient is set based on the particle diameter and cell size. For cells larger than $4d$, a small value of $D = 0.001$ is assigned so that the diffusion-based method degenerates to the PCM. Otherwise, the diffusion coefficient is set as $D = 1.0$.

D. Virtual mass distribution function method

The VMDF is proposed by Zhang, Lu, and Zhang,[28] Ren *et al.*[37] for calculating volume fraction of coarse particles. The VMDF also spreads the influence of coarse particles into the surrounding CFD cells with conservation requirement, where the true mass distribution function is replace by the VMDF with a different particle density. This method is similar to the diffusion-based method and is easy to be implemented and overcomes the deficiencies of the SKM.

E. Big particle method

The concept of the traditional big particle method refers to the DPVM, where the exact portion of volume of particle within the occupied CFD cell is calculated. However, the big particle method does not accurately calculate the portion of volume of particle. It uses an average equalization. As shown in Fig. 4(a), the particle i occupies four cells. The particle volume is divided into four equal portions, and the particle volume fraction in the corresponding cell is calculated similar to PCM (assuming only one particle i in the domain)

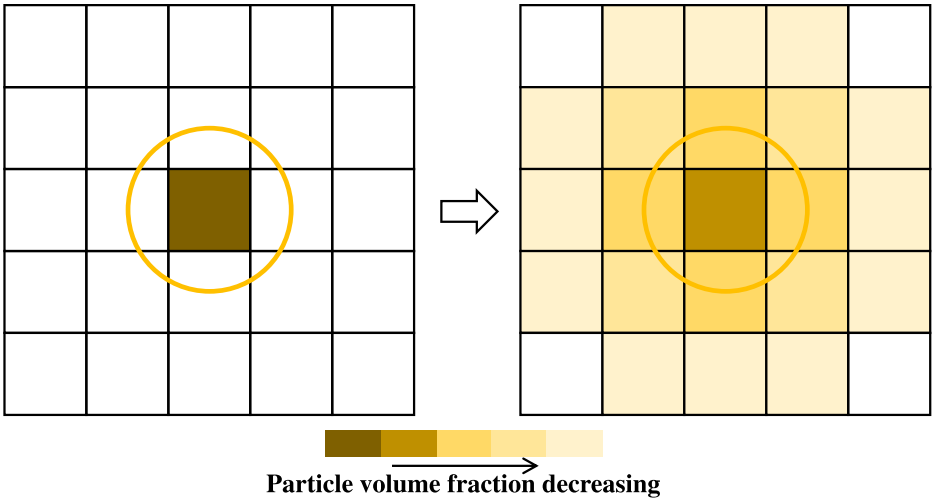

FIG. 3. Schematic representation of the diffusion-based method.

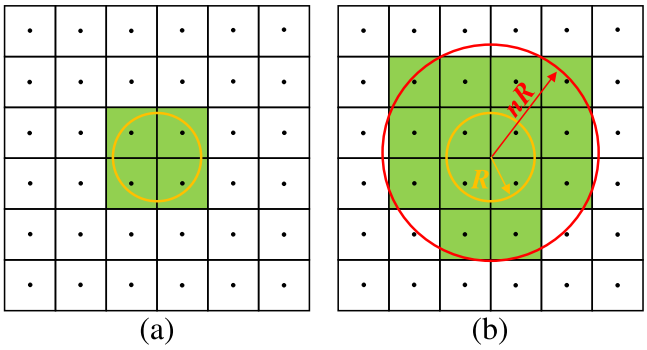

FIG. 4. Schematic representation of the big particle method. The yellow part is the initial particle (a) and the red part is the expanded particle (b). Fourteen cells are occupied by the expanded particle.

$$\alpha_j = \frac{V_p}{V_{c,j}4}. \tag{8}$$

It avoids the determination of the position of the marker points in DPVM. In addition, it is convenient to implement in unstructured meshes. The big particle method also has the drawback of DPVM; that is, it is not applicable when the particle size is close to the cell size.

In this paper, inspired by Gaussian kernel function [Eq. (6)], which diffuses the particle effects to the surrounding CFD cells, we optimize the big particle method by expending the particle in a certain factor. In the Gaussian kernel function, to reduce the computational cost, an important parameter, the expansion size, should be set. The expansion region is a spherical region with the center coinciding with the particle center, whose radius is nR, where n is a scalar factor determining the range of surrounding CFD cells and R is the particle radius. In this paper, based on the Gaussian kernel method, we propose another kernel function as $b(\boldsymbol{x} - \boldsymbol{x}_i) = 1.0$, which is a uniform function. The particle effects are uniformly distributed to the surrounding CFD cells. Figure 4(b) shows the schematic representation of the optimized big particle method. First, the radius of the particle i (R) is expanded by n times (nR). Second, the number of cells (N_p) occupied by the expanded particle is obtained. The linked-list search algorithm[36] is also employed. The expanded particle occupies fourteen cells (green cell) [Fig. 4(b)]. Finally, the particle volume fraction in all the occupied cells is calculated by

$$\alpha_{p,j} = \frac{V_p}{\sum_{j=1}^{N_p} V_{c,j}}. \tag{9}$$

It should be noted that only cells whose center point is within the expanded particle boundary are counted. The value of the expanding factor n will be defined and discussed in Sec. V A.

III. GOVERNING EQUATIONS OF CFD-DEM

The mass and momentum conservation equations of the fluid phase are written as[38,39]

$$\frac{(\partial \rho_f \alpha_f)}{\partial t} + \nabla \cdot (\rho_f \alpha_f \mathbf{u}) = 0, \tag{10}$$

$$\frac{\partial (\rho_f \alpha_f \mathbf{u})}{\partial t} + \nabla \cdot (\rho_f \alpha_f \mathbf{u}\mathbf{u}) = -\alpha_f \nabla p - \mathbf{F}_{\mathrm{pf}} + \nabla \cdot (\alpha_f \rho_f \boldsymbol{\tau}) + \rho_f \alpha_f \mathbf{g}, \tag{11}$$

where α_f is the volume fraction of the fluid phase, $\mathbf{u}$ is the velocity of the fluid phase, ∇p is the pressure gradient, τ is the stress tensor of the fluid phase, and $\mathbf{F}_{\mathrm{pf}}$ is the volumetric two-phase interaction force. Details about $\mathbf{F}_{\mathrm{pf}}$ are given in Appendix A.

The above fluid phase governing equations [Eqs. (10) and (11)] are derived from the point-wise description,[40] where the flow around each individual particle required to be resolved, based on local volume filtering strategy. The characteristic filter size should be larger than the particle; hence, this approach requires for the CFD cell size to be much larger than the particle diameter. This requirement can become problematic under certain circumstances. For example, the particle diameter starts to approach grid size, which is the case of this paper. To alleviate this issue entirely, it is natural to introduce a two-step filtering process.[40] First, the particle data are transferred on the Eulerian mesh through a conservative mollification operation. This filter has a characteristic length scale that corresponds to the mesh size, Δx. Then, the Eulerian field obtained from the first operation is diffused with an operator chosen to lead to the final filtered length scale, which should be much larger than the particle diameter. Based on the two-step filtering process, the governing equations [Eqs. (10) and (11)] are still applicable when the particle diameter starts to approach grid size. The volume fraction calculation methods introduced in Sec. II all meet this requirement.

The motion of the particle has translational and rotational forms, which is governed by Newton's second law and given by

$$m_i \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \mathbf{f}_{\mathrm{pf}} + \sum_{j=1}^{k_c} \mathbf{f}_{c,ij} + m_i \mathbf{g}, \tag{12}$$

$$I_i \frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = \sum_{j=1}^{k_c} \mathbf{M}_{\mathrm{ij}}, \tag{13}$$

where m_i is the particle mass, $\mathbf{f}_{\mathrm{pf}}$ is the fluid–particle interaction force, k_c is the number of particles collision with the particle i, $\mathbf{g}$ is the gravitational acceleration, and $\mathbf{f}_{c,ij}$ is the collision force. $\boldsymbol{\omega}_i$ and I_i are the angular velocity, and moment of inertia of the particle. $\mathbf{M}_{\mathrm{ij}}$ is the torque due to the collision of particle/particle or particle/wall. One of the soft sphere models, the linear spring–dashpot model, is used to calculate the collision force.[41] More details about governing equation of the CFD-DEM can be found in Ref. 28.

IV. MODEL IMPLEMENTATION

The governing equations of the fluid phase Eqs. (10) and (11) are discretized by the finite volume method (FVM) using the open-source code OpenFOAM. It is assumed that the fluid phase is incompressible in this paper. Extracting the density item from Eqs. (10) and (11), the mass and momentum conservation equations of the fluid phase can be rewritten as (see Appendix B)

$$\frac{\partial \alpha_f}{\partial t} + \nabla \cdot (\alpha_f \mathbf{u}) = 0, \tag{14}$$

$$\frac{\partial}{\partial t}(\alpha_f \mathbf{u}) + \nabla \cdot (\alpha_f \mathbf{u}\mathbf{u}) - \mathbf{u}\left(\frac{\partial \alpha_f}{\partial t} + \nabla \cdot (\alpha_f \mathbf{u})\right) - \nabla \cdot (\alpha_f \boldsymbol{\tau}) = -\alpha_f \nabla \frac{p}{\rho_f} - \frac{\mathbf{F}_{\mathrm{pf}}}{\rho_f} + \alpha_f \mathbf{g}. \tag{15}$$

The third term on the left-hand side of Eq. (15) is retained to maintain the boundedness of the solution variable and promotes better convergence.[42] The semi-discretized algebraic equation of Eq. (15) is

$$a_p \mathbf{u}_p + \sum a_n \mathbf{u}_n = s_p - \left(\alpha_c \nabla \frac{p}{\rho_f}\right)_p + \left(-\frac{\mathbf{F}_{pf}}{\rho_f} + \alpha_f \mathbf{g}\right)_p, \quad (16)$$

where a_p and a_n are the diagonal and off-diagonal elements of the coefficients matrix, respectively, $\mathbf{u}_p$ is the fluid velocity of the cell p, $\mathbf{u}_n$ is the fluid velocity of the adjacent cell, and s_p is the source term. The discretized form of the pressure gradient is not included in Eq. (16) for preventing the pressure oscillations in the spirit of the Rhie–Chow interpolation.[43] According to Zhang, Zhao, and Bayyuk,[44] the body force term is also not included in s_p. In addition, the volumetric two-phase interaction force is explicitly discretized because of coarse particles. Equation (16) can then be written as

$$\mathbf{u}_p = \frac{1}{a_p}\left(-\sum a_n \mathbf{u}_n + s_p\right) - \frac{\alpha_{f,p}}{a_p}\left(\nabla \frac{p}{\rho_f}\right)_p + \frac{1}{a_p}\left(-\frac{\mathbf{F}_{pf}}{\rho_f} + \alpha_f \mathbf{g}\right)_p. \quad (17)$$

The fluid velocity $\mathbf{u}_p$ does not satisfy the continuity constraint. Hence, a pressure Poisson equation is given to correct the fluid velocity. We define a face field **HbyA**, which represents the finite volume representation of the spatial convective and diffusive fluxes of the phase momentum,[45] as

$$\mathbf{HbyA} = \left[\frac{1}{a_p}\left(-\sum a_n \mathbf{u}_n + s_p\right)\right]_f, \quad (18)$$

where subscript f represents the quantity on the cell surface. Integrating Eq. (14) over a finite volume cell and substituting Eq. (17) into Eq. (14), the pressure Poisson equation can be constructed as follows:[46,47]

$$\left(\frac{\alpha_f^2}{a_p}\nabla \frac{p}{\rho_f}\right)_f \cdot \mathbf{S}_f = \left(\frac{\partial \alpha_f}{\partial t}\right)_f + \left(\alpha_f \mathbf{HbyA} - \frac{\alpha_f}{a_p}\frac{\mathbf{F}_{pf}}{\rho_f} + \frac{\alpha_f^2 \mathbf{g}}{a_p}\right)_f \cdot \mathbf{S}_f, \quad (19)$$

where $\mathbf{S}_f$ represents the exterior normal vector of the cell face. Equation (19) is different from the basic pressure Poisson equation, which is motivated by numerical stability considerations. In OpenFOAM, except for the boundary face, all variables are stored at the cell centroid. The Rhie–Chow interpolation is adopted to avoid pressure checker-boarding.[43] The convection and diffusion terms is integrated by the Gauss theorem with a second-order interpolation scheme. The gradient term is solved using the linear face interpolation method with corrections to the non-orthogonality of the mesh. A first-order implicit Euler scheme is used for the temporal discretization. The time step for the fluid phase is set based on the Courant–Friedrich–Lewy (CFL) criteria where the CFL number is 0.5 to ensure convergence. The open-source code LIGGGHTS is used to solve the particle motion and inter-particle collisions. The time steps for the particle phase are chosen in order to keep it less than 10% the Rayleigh time[48]

$$t_R = \frac{0.5\pi d}{0.163\nu + 0.8766}\sqrt{\frac{2\rho_p(1+\nu)}{Y}}, \quad (20)$$

where ρ_p is the particle density, ν represents the Poisson's ratio, and Y denotes the Young's modulus.

For the particulate flows with coarse particles, an important concern is how to estimate the background fluid velocity and particle volume fraction, which will be used to calculate the drag force.[32,49] In this paper, we use the optimized big particle method for estimating the background fluid velocity and particle volume fraction from the surrounding cells, as shown in Fig. 4. Then, the background fluid velocity is calculated as follows:

$$\mathbf{u} = \frac{\sum_{j=1}^{N} \mathbf{u}(\mathbf{r}_j) V_{c,j}}{\sum_{j=1}^{N} V_{c,j}}, \quad (21)$$

where j is the index of cell j with a corresponding volume $V_{c,j}$, and $\mathbf{u}(\mathbf{r}_j)$ is the fluid velocity in cell j. In this paper, the expanding factor n of the optimized big particle method is set as 5.0, which will be discussed in Sec. V A. Similarly, the corrected particle volume fraction for drag force calculation is calculated as

$$\bar{\alpha}_p = \frac{\sum_{j=1}^{N_p} \alpha_{p,j} V_{c,j}}{\sum_{j=1}^{N_p} V_{c,j}}, \quad (22)$$

where $\alpha_{p,j}$ is the particle volume fraction in cell j.

V. NUMERICAL SIMULATIONS AND COMPARISONS

For accessing the performance of different methods for calculating the volume fraction of coarse particles, three test cases are presented. The first one is a particle distribution case. The second one is a monodisperse gas–solid fluidized bed. The last one is a binary water–solid fluidized bed case. In the first case, our focus is solely on the particle volume fraction field, disregarding the movement of particles. This approach allows us to evaluate the efficacy of different methods in accurately calculating the particle volume fraction. In the second and third cases, the interactions between particles and particles, as well as particles and fluids, are to assess the effectiveness of different methods in determining the instantaneous particle volume fraction and the resulting distribution of particles. The polydisperse effects are considered in the third case, compared to the second case.

A. Particle distribution

The particle distribution case is conducted in this section, as shown in Fig. 5. A domain with three dimensions, measuring $160d$ in length, $160d$ in width, and $4d$ in thickness, is employed. The thickness is set as $4d$ to ensure that the CFD cell is a regular hexahedron. Particles are located in the middle of the domain with a size of $60d \times 60d \times 4d$. The particle numbers are calculated from the average particle volume fraction (α_p),

$$N_p = \frac{60d \times 60d \times 4d \times \alpha_p}{\frac{1}{6}\pi d^3} = \frac{86\,400\alpha_p}{\pi}. \quad (23)$$

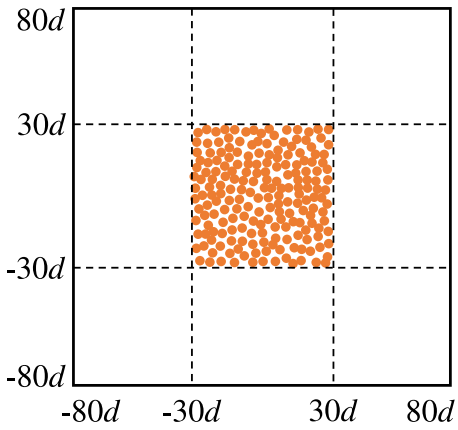

FIG. 5. Diagram of the particle distribution with a size of $60\,d \times 60\,d \times 4d$ in the middle region.

The average particle volume fraction is 0.3 in this case. Since the purpose is to compare the particle volume fraction calculated by different methods, the two-phase interactions, particle collisions, particle gravity, etc., are not considered. The initial velocity of the two-phase is zero. The boundary conditions for the velocity and pressure of the fluid phase at the four boundaries are zero gradient.

Five cell sizes are used with the ratio of cell size to particle size $\Delta x/d$ ranging from 0.5 to 2.0, as shown in Table I. The N_x, N_y, and N_z are the number cells in the x, y, and z directions, respectively. The cell size is computed by $\Delta x = V_c^{1/3}$. Figure 6 gives the contour of particle volume fraction based on different methods, including the DPVM, the two-grid method, the big particle method, and the diffusion-based method. $\Delta x/d$ is set as 1.0. Each cell is colored according to the corresponding α_p value in the cell. Because the particle size distribution is monodisperse, the limiting particle volume fraction is 0.63. As is evident from the contours, some high or low values (thus large gradients) are frequent in the DPVM results, also occasionally occur in the two-grid method and big particle methods result. No such large values are present in the contour obtained with the diffusion-based method.

TABLE I. The details of the cell sizes for investigating the different methods.

Mesh	N_x	N_y	N_z	$\Delta x/d$
M1	80	80	2	2.0
M2	160	160	4	1.0
M3	239	239	6	0.67
M4	320	320	8	0.5

Figure 7 shows the particle volume fraction in line $y = 0$. The maximum value of y axis is 0.63, which is the limiting particle volume fraction. At $\Delta x/d = 2.0$, the particle volume fraction based on DPVM is close to the average value, which is set as 0.3 initially. However, when $\Delta x/d$ tends to the value of 1.0, the particle volume fraction in line $y = 0$ oscillates violently [Fig. 7(a)]. The maximum particle volume fraction can exceed 0.6, and the minimum value is less than 0.1. The characteristics of the oscillation distribution may have an important influence on the simulation results of CFD-DEM. Therefore, the DPVM may be inapplicable when the particle size is equal to or greater than the cell size.

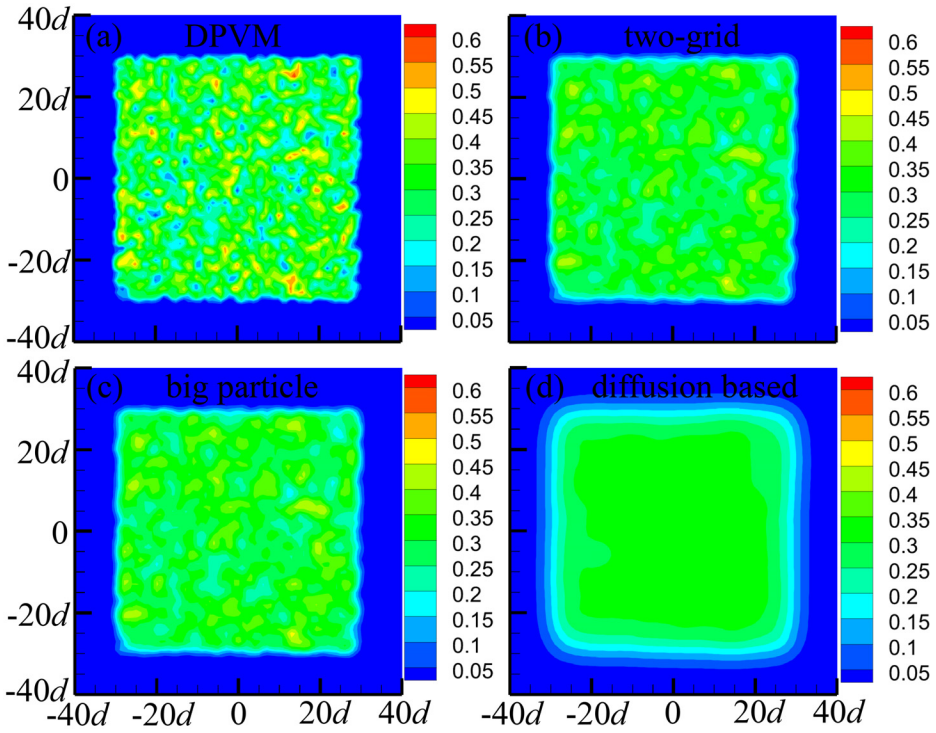

FIG. 6. The particle volume fraction field based on, (a) DPVM, (b) two-grid method, (c) big particle method, and (d) diffusion-based method, with $\Delta x/d = 1.0$.

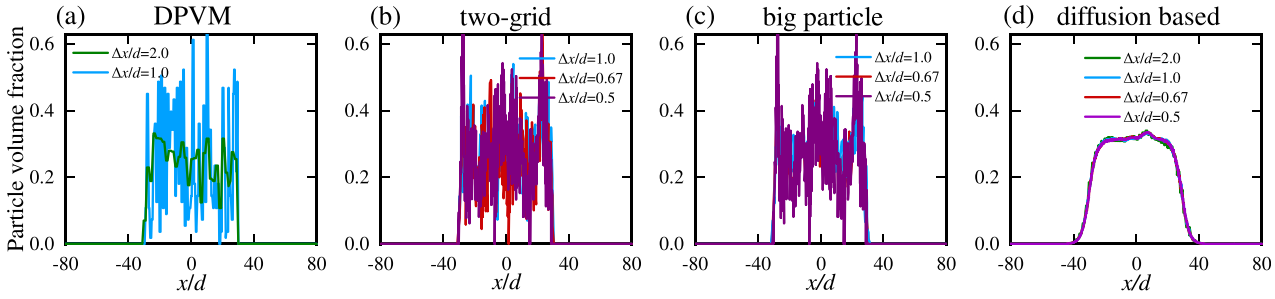

FIG. 7. The particle volume fraction calculated from different methods, (a) DPVM, (b) two-grid method, (c) big particle method, and (d) diffusion-based method.

Figures 7(b) and 7(c) show the results of the two-grid method and the big particle method, respectively. According to the discussion in Sec. II, the coarse CFD cell size determined based on the particle size in the two-grid method to ensure that the ratio between the coarse cell size and the particle size is greater than 3.0. In this paper, the simplified two-grid method is used to avoid data exchange between the CFD mesh and the particle mesh. In general, the simplified two-grid method is similar to the big particle method with the expanding factor of $n = 3$. As a result, the particle volume fraction in line $y = 0$ obtained from the two-grid method and the big particle method shows similar trend. Some high and low values are also frequently observed in the results of the two-grid and big particle methods. This may be attributed to the small value of the expanding factor, n. Figure 7(d) is the simulation results of the diffusion-based method, where the particle volume fraction curve is smooth and the values are close to the average volume fraction for all four size ratios. The comparison shows that the diffusion-based method has the widest application range and provides the most accurate calculation results. For quantitative comparison of particle volume fraction distributions, Fig. 8 gives the histograms of the particle volume fraction distributions calculated from different methods. The ratio of cell size to particle size is $\Delta x/d = 1.0$. It can be seen in the figure that the particle volume fraction obtained from the DPVM is totally different from the average particle volume fraction because maximum probability density occurs at approximately $\alpha_p = 0.05$. In addition, the probability density around $\alpha_p = 0.6$ is also comparable to the small particle volume fraction. The maximum probability density of the particle volume fraction based on the two-grid method, the big particle method, and the diffusion-based method occurs at $\alpha_p = 0.3$. However, the particle volume fraction of the two-grid and the big particle methods is distributing between 0 and 0.63, while it distributes between 0 and 0.35 for the diffusion-based method.

In the above simulation, the particle radius expands to three times of the initial particle radius in the two-grid method and four times in the big particle method. However, the appropriateness of $n = 3.0$ or 4.0 requires further discussion. Figure 9 shows the particle volume fraction calculated based on the big particle method with different expanding factors. For example, $n = 2.0$ means twice the particle radius. The size ratio is set as $\Delta x/d = 0.67$. When the particle radius is expanded to two times or three times the initial particle radius in the big particle method, the maximum particle volume fraction is larger than 0.6, which is much greater than the average particle volume fraction. Furthermore, there is a significant fluctuation observed in the particle volume fraction field. However, the particle volume fraction curve becomes stable when the particle radius expands to four times and five times. As a result, the expanding factor n should be set at four or five in the optimized big particle method or the simplified two-grid method.

Since the big particle method needs to search for the number of cells occupied by the expanded particles, the number of cells obtained in the boundary region and the middle region of the domain may be different, which will result in different particle volume fraction. To discuss the effect of the expanding size of the big particle method on the calculation of the particle volume fraction at the boundary region, the particles are initially distributed in the near-wall region. Then, the particle volume fraction at the line $y = 0$ is shown in Fig. 10 with different expanding factors n. The particle volume fraction at the near boundary

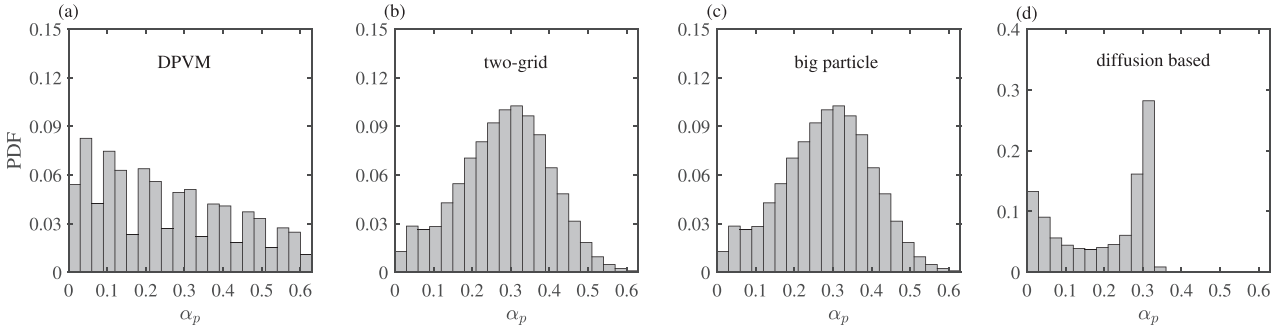

FIG. 8. The histograms of the particle volume fraction distributions calculated from different methods, (a) DPVM, (b) two-grid method, (c) big particle method, and (d) diffusion-based method. The ratio of cell size to particle size is $\Delta x/d = 1.0$.

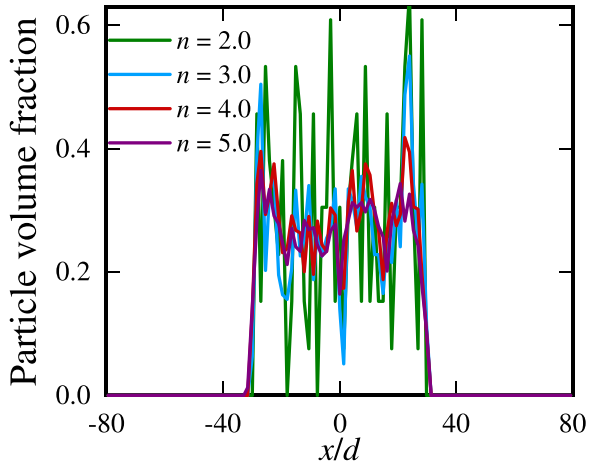

FIG. 9. The particle volume fraction calculated based on the big particle method with different expanding factors.

is greater than that far from the boundary. It will lead to an inaccurate calculation of the two-phase interaction force at the boundary, which will be discussed in Sec. V B.

B. Monodisperse gas–solid fluidized bed

In this section, a monodisperse gas–solid fluidized bed case is used to compare the simulation results with different methods for calculating the particle volume fraction. The gas–particle and particle–particle interactions are considered. Figure 11 shows the computation domain with size of $10 \times 44 \times 120\,\text{mm}^3$ in three directions, which references the experiment of Müller *et al.*[50,51] The particle diameter and density are 1.2 mm and $1000\,\text{kg/m}^3$, respectively. At the initial state, approximately 9240 particles are free to accumulate at the bottom, and the stack height is 30 mm, as shown in Fig. 11. Thereafter, a constant flux of gas is introduced from the bottom to fluidize the

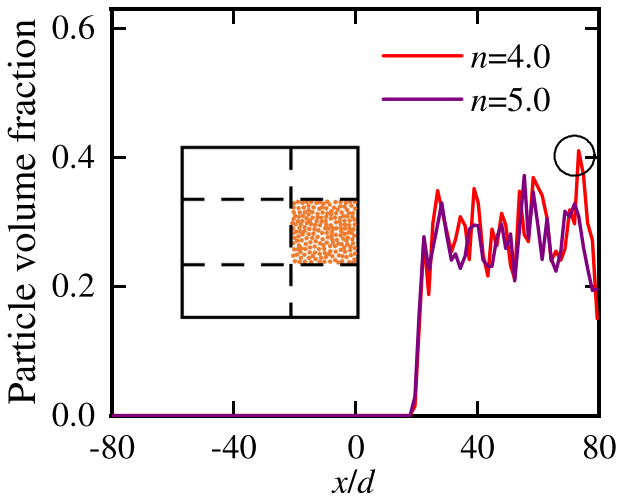

FIG. 10. The particle volume fraction in the boundary region based on the optimized big particle method.

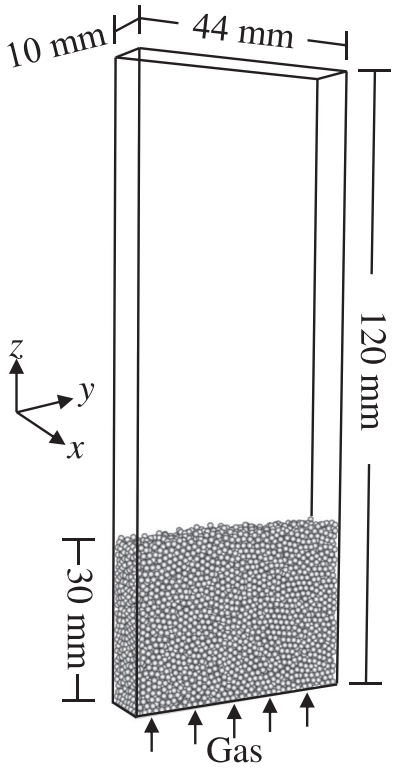

FIG. 11. The computation domain of the monodisperse gas–solid fluidized bed with size of $10 \times 44 \times 120\,\text{mm}^3$ in the *x*, *y*, and *z* directions, respectively. The initial particle bed height is 30 mm.

packed particles. Müller *et al.*[50,51] used magnetic resonance to measure the time-averaged gas volume fraction and the particle velocity. Therefore, the simulation results can be compared to the experimental measurement. All the parameters used in this simulation case are given in Table II.

The pressure boundary condition on the upper *z* and lower *z* face is set as a fixed value and zero gradient, respectively. The gas velocity is set as zero gradient on the upper *z* face and fixed value of 0.9 m/s on the lower *z* face. At the four side faces, the gas velocity is set as zero value and the pressure is set as zero gradient. Initially, the velocity and pressure of the gas are zero in the interior domain and the velocity of the particles is also zero. The time-averaged gas volume fraction at the heights of $z = 16.4$ and 31.2 mm and particle vertical velocity at the heights of $z = 15$ and 25 mm are extracted from the numerical simulation since the experimental measurement data only covered these four sections. The simulation first runs for 10 s, and then, the time averaging starts to ensure that the calculated averages are free from the initial condition and associated startup transient, which means that the total duration of each case is 10 s plus the averaging time. We set different averaging times in the CFD-DEM simulations, 12, 16, and 20 s, respectively, to investigate the effect of averaging time on the time-averaged results. It is found that statistical convergence is achieved at an averaging time of approximately 16 s. As a result, we set the averaging time as 20 s in the numerical simulation, which is considered sufficiently long to obtain statistically time-averaged results. The three methods for

TABLE II. All parameters used in the monodisperse gas–solid fluidized bed case. The sliding friction coefficient is used when calculating the normal collision force between particles. Details about particle collisions can be found in Ref. 52.

Parameters	Values
Geometry of the fluidized bed	
Width, length, height (mm)	10, 44, 120
Solid properties	
Density	1000 kg/m^3
Diameter	1.2 mm
Number	9240
Young's modulus	1.0×10^7 Pa
Poisson's ratio	0.33
Coefficient of restitution	0.90
Sliding friction coefficient	0.10
Gas properties	
Density	1.2 kg/m^3
Viscosity	1.5×10^{-5} m^2/s
Inlet velocity	0.9 m/s

calculating the particle volume fraction are also applied in this case to compare their performance. With the exception of the particle volume fraction calculation method, all other parameters and boundary conditions are the same. The cell size is set as $N_x \times N_y \times N_z = 8 \times 36 \times 100$, with the ratio of cell size to particle size $\Delta x/d = 1.0$. The expanding factor n of the big particle method is set as 5.0 according to the simulation results in Sec. V A. Both the simulation results of the optimized big particle method and the diffusion-based method appear to be consistent with the experiment. The different particle behavior of the two methods may be due to the different instantaneous particle volume fraction, which significantly affects the gas–particle interactions. However, after time averaging, the simulation results of the two methods are similar, which is shown below.

First, the gas volume fraction obtained from different methods is compared with the experimental measurement data. Figure 12 shows the contours of particle volume fraction to illustrate the bubble dynamics in the fluidized bed. The bubbles can be clearly identified from the contours of the gas volume fraction based on the big particle method and the diffusion-based method. However, in the DPVM results shown in Fig. 12(a), the bubbles are not identified, which is due to the large local variations in the gas volume fraction field. From Figs. 12(b) and 12(c), the bubble formation ($t = 0.55$ and 0.65 s), growth (0.65 and 0.70 s), and burst (0.70 and 0.80 s) can be clearly identified. The bubble dynamics observed in the numerical simulation are physically reasonable as confirmed in previous experiments.[50]

Figure 13 gives the time-averaged gas volume fraction at two different heights, $z = 16.4$ and 31.2 mm based on different methods. Although when $\Delta x/d = 1.0$, the instantaneous particle volume fraction calculated on the basis of the DPVM has a significant error [Fig. 7(a)], the oscillation characteristics of the volume fraction disappear after a long time average. The average gas volume fraction of the DPVM is generally consistent with the experimental measurement data. The gas volume fraction calculated based on the big particle method also agrees well with the experimentally measured value along the y direction at the height of $z = 16.4$ mm [Figs. 13(a) and 13(c)]. It should be noted that the gas volume fraction near the wall is larger than the experimental measurement. This phenomenon is particularly evident at the height of $z = 31.2$ mm. For the big particle method, this is due to that the number of cells occupied by the particle by expanding the particle size near the wall is less than that in the middle region. Then, the particle volume fraction near the wall will increase due to the decrease in searching cell numbers. It then increases the drag force on the particles. For the diffusion-based method, Müller *et al.*[51] attributed the phenomenon to the over-prediction of the bubbles width in the CFD-DEM. It is worth noting that the instantaneous and time-averaged volume fraction calculated by the diffusion-based method are smooth, and the calculation results are accurate. The main reason is that the diffusion-based method obtains the particle volume fraction field by solving the volume fraction diffusion equation, which on the one hand ensures the conservation and on the other hand makes the volume fraction field smooth. This treatment is very effective in dense particulate flows. Comparing Figs. 13(a) and 13(c), we find that the gas volume fraction is basically unchanged for the diffusion-based method and the optimized big particle method, which proves the wide application range of these two methods.

The time-averaged particle vertical velocity obtained from different methods at the height of $z = 15$ and 25 mm is shown in Fig. 14. In general, the particle vertical velocities calculated by the three methods are basically agreeing with the experimentally measured data. It can be found that the particle vertical velocity is larger than the experimental data based on the big particle method, which is obvious at the height of $z = 25$ mm. This is because the instantaneous particle volume fraction field calculated from the big particle method is not smooth, which makes the particle volume fraction in a certain cell may be very large or small. Then, the drag force on the particles increases or decreases dramatically with the change of the particle volume fraction because the drag force is related to the instantaneous particle volume fraction. At the height of $z = 25$ mm, the particle vertical velocity obtained from the diffusion-based method is smaller than the experimental data. In general, the big particle method and the diffusion-based method all perform well. However, the simulation results obtained from the optimized big particle method are closer to the experimental value.

C. Binary water–solid fluidized bed

The above cases only involve single size particles. However, particulate flows in industry usually involve coarse particles of different sizes. In this section, we simulate a binary water–solid fluidized bed where the CFD cell sizes approach the sizes of the large particles and are three times larger than the small particles to see the utility of using a revised particle volume calculation. The geometry of the simulation domain is a square tube with dimension $0.01 \times 0.01 \times 0.12$ m^3 in the x, y, and z directions, respectively. The particles are uniformly distributed in the tube initially with the diameter of big particles and small particles of 1 and 0.3 mm, respectively. The numbers of the big and small particles are 42 441 and 1146. The water is uniformly injected from the bottom with a constant superficial velocity of 0.01, 0.015, 0.02, and 0.025 m/s. The particle density is 2000 kg/m^3, while the water density is 1000 kg/m^3 with the viscosity of 1.0×10^{-6} m^2/s. The computational domain is divided into $10 \times 10 \times 120$ structured mesh cells, where the ratio of cell size to particle size $\Delta x/d$ is 1.0 and 3.3 for the big particles and small particles, respectively. Different from the gas–solid fluidized bed case, the lubrication force is important in

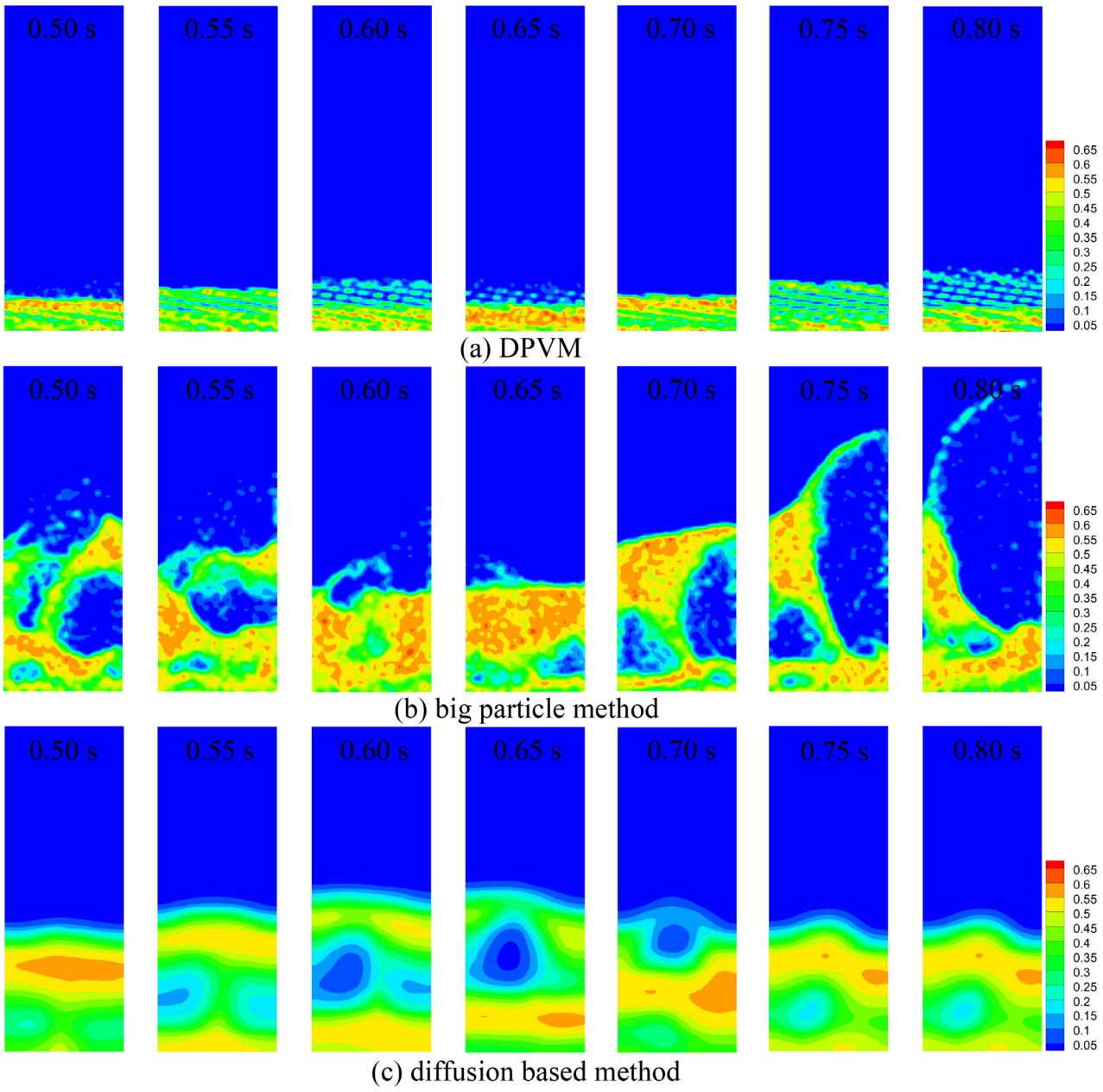

FIG. 12. Contours of particle volume fraction during a cycle of bubble formation and evolution obtained from (a) DPVM, (b) big particle method, and (c) diffusion-based method.

water–solid case. For dense water–particle flow simulation, lubrication force, which results from radial pressure of fluid when two particles get close and the interstitial fluid is squeezed out, is also an important mechanism. The particle collisions immersed in fluid are more inelastic than dry particle collisions. In this paper, the mesh is not fine enough to resolve the lubrication effect. It can be considered by introducing a relatively small restitution coefficient, i.e., $e = 0.1$, for particle–particle and particle–wall collisions for simplicity. This approximate approach is applied by other researchers for simulating fluid–particle interactions.[53–55] All the parameters used in this simulation case are shown in Table III. We simulate this case with different methods for calculating the particle volume fraction. It was found that the mixed binary particles will eventually separate each other with the small particles on the top and large particles on the bottom.[56] As a result, we use the drag law of monodisperse particles.

We can estimate the particle volume fraction by using the Richardson and Zaki relation. Richardson and Zaki[57] proposed a well-known empirical relationship between the superficial fluid velocity and

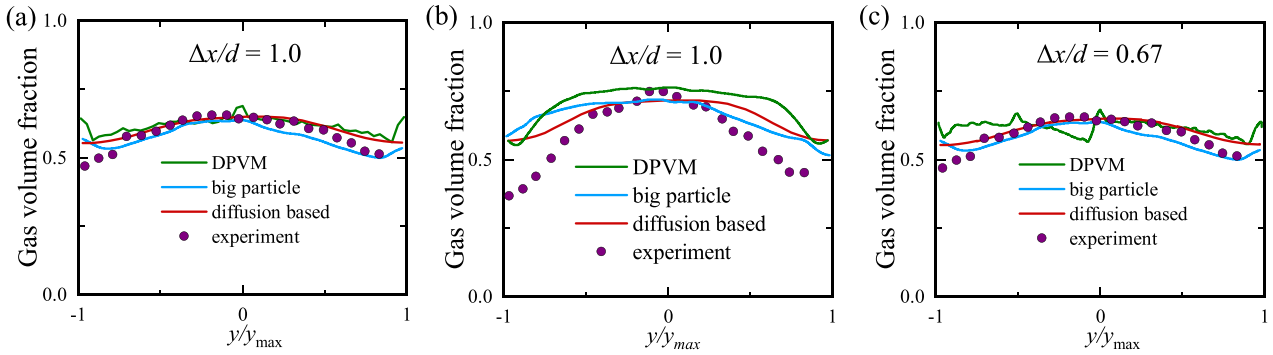

FIG. 13. The time-averaged results obtained from different methods for calculating the particle volume fraction with (a) gas volume fraction at the height of $z = 16.4$ mm and $\Delta x/d = 1.0$, (b) gas volume fraction at the height of $z = 31.2$ mm and $\Delta x/d = 0.67$, and (c) gas volume fraction at the height of $z = 16.4$ mm and $\Delta x/d = 1.0$. The experimental measurement data are also given for comparison.

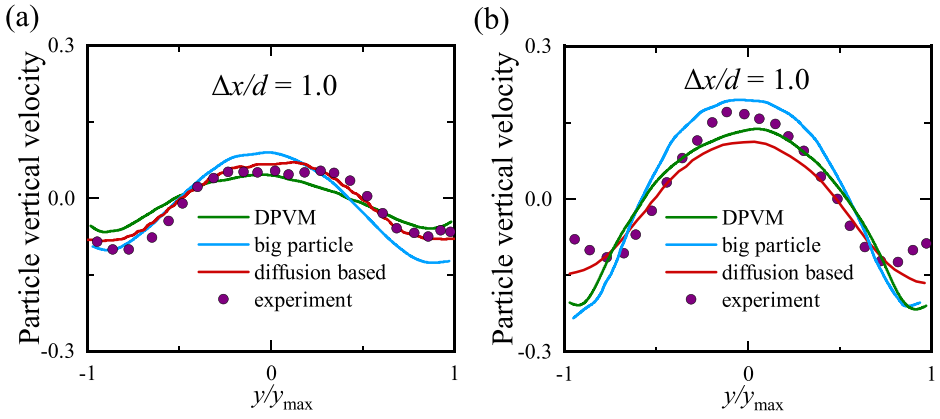

FIG. 14. The time-averaged results obtained from different methods for calculating the particle volume fraction with (a) particle vertical velocity at the height of $z = 15$ mm and $\Delta x/d = 1.0$, and (b) particle vertical velocity at the height of $z = 25$ mm and $\Delta x/d = 1.0$. The experimental measurement data are also given for comparison.

TABLE III. All parameters used in the binary water–solid fluidized bed case.

Parameters	Values
Geometry of the square tube	
Length, width, height (mm)	10, 10, 120
Solid properties	
Density	2000 kg/m^3
Diameter (mm)	0.3 and 1
Number	42 441 (0.3 mm) and 1146 (1 mm)
Young's modulus	1.0×10^8 Pa
Poisson's ratio	0.23
Coefficient of restitution	0.10
Sliding friction coefficient	0.10
Water properties	
Density	1000 kg/m^3
Viscosity	1.0×10^{-6}m^2/s
Inlet velocity	0.01, 0.015, 0.02, 0.025 m/s

the particle volume fraction based on experiment, which is expressed as

$$\frac{u}{w_t} = (1 - \alpha_p)^n, \tag{24}$$

where u is the superficial fluid velocity, and w_t is the terminal settling velocity of a single particle in an infinitely large domain, which can be calculated with $Re_t = (2.33Ar^{0.018} - 1.53Ar^{-0.016})^{13.3}$,[58] where $Re_t = w_t d/\nu_f$ is the particle Reynolds number, $Ar = (s-1)d^3 g/\nu_f^2$ is the Archimedes number, d is particle diameter, ν_f is the kinematic viscosity of the fluid, and $s = \rho_p/\rho_f$ is the density ratio between the particle and fluid. The exponent n depends on Re_t, which can be calculated by a correlation $(5.1 - n)/(n - 2.7) = 0.1Re_t^{0.9}$.[59] Then, based on the formula (24), we can obtain the particle volume fraction of the small and big particles.

Figure 15 shows a qualitative comparison of the particle distribution under different inlet velocities ($u = 0.01$, 0.015, and 0.02 m/s) at the time of 20 s. In each subfigure, the simulation results are obtained based on the diffusion-based method, the big particle method, and DPVM from left to right. Figure 15 clearly shows differences in the

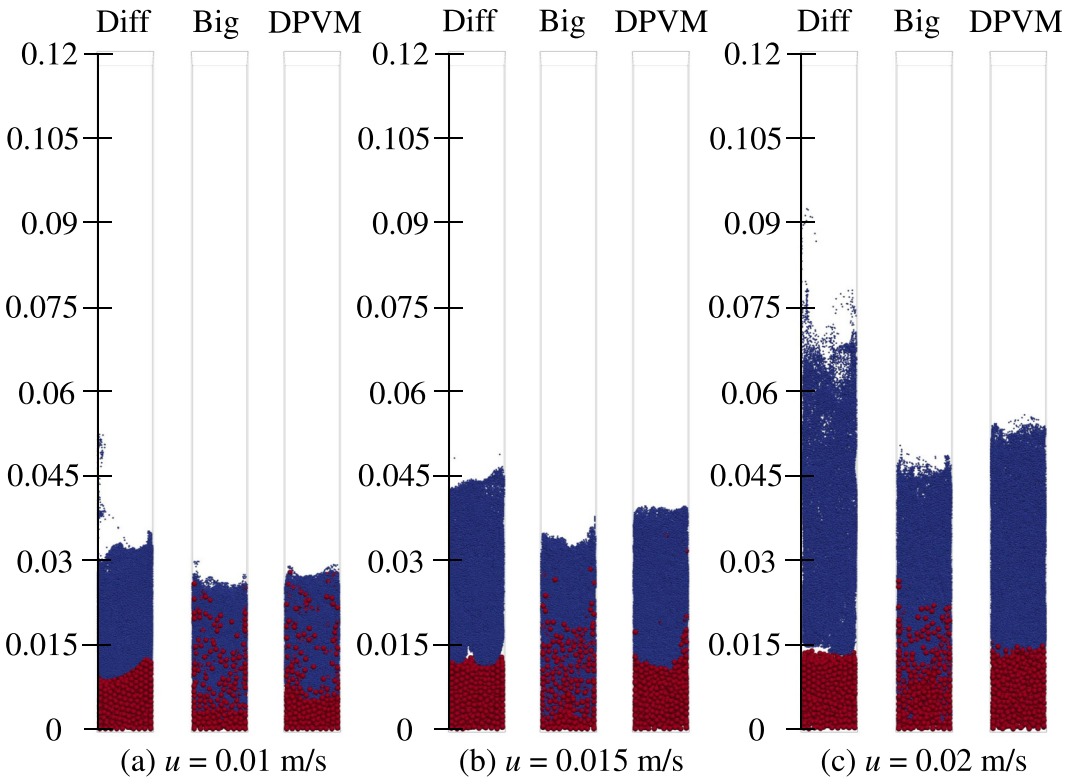

FIG. 15. The particle distribution under different inlet superficial fluid velocities (a) $u = 0.01$ m/s, (b) $u = 0.015$ m/s, and (c) $u = 0.02$ m/s at the time of 20 s. In every subfigure, the simulation results are obtained based on the diffusion-based method (abbreviated as Diff in the figure), the big particle method (abbreviated as Big in the figure), and DPVM from left to right.

fluidization behavior depending on the particle volume fraction methods. The small and big particles are clearly separated with no mixing region for the diffusion-based method. However, the small and big particles are mixed each other at the interface for the simulation results based on the big particle method and the DPVM. The size of the mixing region obtained from the big particle method is larger than that of the DPVM. The difference in the size of the mixing region leads to different fluidization heights of the particles. Actually, according to the experiment of Khan *et al.*,[56] the small and big particles will mix with each other at their interface after separation. The simulation results based on the big particle method and the DPVM are more consistent with the experimental phenomena than the diffusion-based method. It may be due to the fact that the diffusion-based method does not consider the polydisperse effects when calculating the particle volume fraction because it does not consider the variation in particle sizes and sets the same diffusion coefficient for both small and big particles. It results in an even smoothing of exchanged information across all particles with different sizes.[60] The polydisperse effects are very important at the interface between the small and big particles. In this paper, we optimize the big particle method by referring the statistical kernel method to diffuse the coarse particles effects to the surrounding CFD cells. The particle is expended in a certain factor n and the expansion region is a spherical region with the center coinciding with the particle center, whose radius is nR. The size of the expansion region is related to the particle size. For particles of varying sizes, we obtain different expansion regions, which mean that the effects of particle size (polydisperse effects) are taken into account when calculating the particle volume fraction field. For the small particles in this simulation case where the cell size is three times larger than the small particle size, even we expand the small particle size by three times, the particle volume fraction calculated is still same to the PCM because the expanded particle is smaller than the cell size. The optimized big particle method has no effect on the volume fraction calculation for small particles. In addition, the DPVM also consider the polydisperse effects due to exact portion of particle in each cell are determined. However, as discussed above, the DPVM is not applicable when the particle size approaches the cell size. As a result, the optimized big particle method proposed in this paper is the best choice for the binary water–solid fluidized bed.

Similarly, Fig. 16 shows the particle volume fraction at the central axis line ($x = 0$ and $y = 0$) under different inlet velocities at the time of 20 s. The black dotted lines in the figure represent the predicted average volume fraction of small (line in the bottom) and big particles (line in the top) based on the Richardson and Zaki relation given in Eq. (24). It can be seen in the figure that all the three methods (diffusion-based, big particle, and DPVM) give accurate volume fraction of the big particles. For example, in Figs. 16(b)–16(d), the particle volume fraction in the big particle region is around the predicted line (line in the top). Some high or low values are frequently observed in the DPVM results due to the big particle size approaching the cell size, which is similar to the discussion in Sec. V A. At the small particle region, the calculated

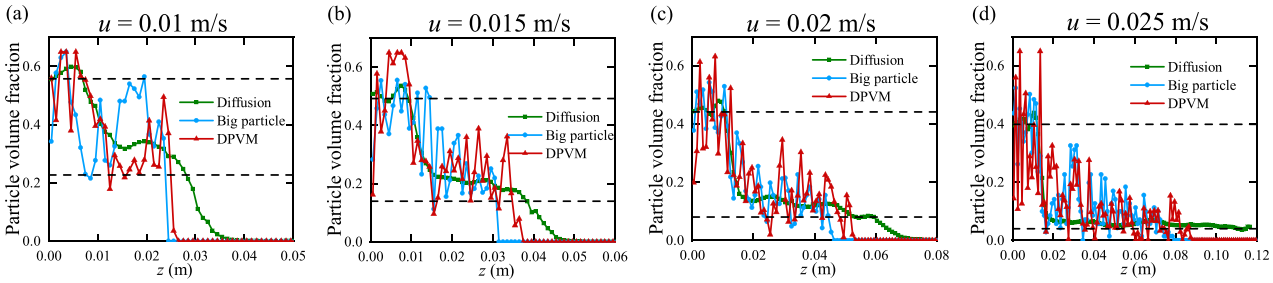

FIG. 16. The particle volume fraction at the central axis line ($x = 0$ and $y = 0$) under different inlet superficial fluid velocities (a) $u = 0.01$ m/s, (b) $u = 0.015$ m/s, (c) $u = 0.02$ m/s, and (d) $u = 0.025$ m/s at the time of 20 s. The black dotted lines in the figure are the predicted average volume fraction of the small (line in the bottom) and big particles (line in the top) based on the Richardson and Zaki relation.

particle volume fraction is clearly larger than the predicted value. We use the inlet velocity at the bottom to predict the small particle volume fraction. However, the water needs to pass through the big particles first, which may affect the distribution of small particles. Abrupt change on the particle volume fraction line can be found for the diffusion-based method, which shows the interface of the small and big particles. For the results of the big particle method and the DPVM, we can observe the mixing region from the high particle volume fraction at the interface.

At last, we discuss the computation costs of different methods. The computational costs of CFD-DEM can be divided into four parts: cost of solving the fluid phase governing equations, cost of solving the DEM (particles), cost induced by the complicated solid quantity calculations, and cost induced by the fluid phase reconstruction (background fluid velocity, background particle volume fraction, etc.). The most expensive part of the simulations is associated with solving the DEM model such as the particle position updates and collision detection. We use the binary water–solid fluidized bed case as an example, where 43 578 particles are tracked. The simulation cost for the DEM model is about 23 times as much the cost associated with solving the water phase governing equations. We use the CPU time spent on solving the water phase governing equations as a reference for each case. For different particle fraction calculation methods, the solid quantity calculation costs are computed as 4.82 (diffusion based), 4.37 (big particle), 4.91 (DPVM), and 3.85 (PCM), respectively. The costs induced by the fluid phase reconstruction are calculated about 2.69. As a result, after using these methods, the total computation costs increase by no more than 5% compared with the PCM. The big particle method needs less costs than the diffusion-based method. We then give the comparison between different methods in Table IV, such as the application range, the computation cost, etc., to show the advantages and disadvantages of these methods.

VI. SUMMARY AND CONCLUSION

The particle volume fraction is a vital factor that greatly affects the accuracy of CFD-DEM simulations. In some situations, the particle size is large. It may result in particle size comparable to or larger than the CFD cell size. Therefore, the methods for calculating the volume fraction of coarse particles were given by some researchers, such as the DPVM, the two-grid method, the big particle method, and the diffusion-based method.

In this paper, we thoroughly investigate the performance of different methods for computing volume fraction of coarse particles with different particle sizes and CFD cell sizes. The different methods, such as the DPVM, the two-grid method, etc., are reviewed by introducing their origin, principle, and disadvantages. The two-grid method uses two independent meshes for the coarse particles and the CFD simulation. We simplify the two-grid method to avoid the data exchange between the CFD mesh and particle mesh. The big particle method is similar to the DPVM. We optimize it by expanding the particle size to obtain the number of cells occupied by the particle. Then, the particle volume is uniformly distributed to the occupied cells. The diffusion-based method distributes the particle effects to the surrounding CFD cells by solving a transient, homogeneous diffusion equation. We use a variable diffusion coefficient in this paper to make the diffusion-based method suitable for different particle sizes. All the above methods are implemented in the open-source code CFDEM.

Three numerical tests were conducted to compare the simulation results obtained from different methods. These tests include a stationary particle distribution case, a monodisperse gas–solid fluidized bed case, and a binary water–solid fluidized bed case. Extreme high or low values of the instantaneous particle volume fraction are frequent in the DPVM results when $d/\Delta x > 1.0$ with the largest value being approximately 0.6, which is close to the particle packing limit. As a result, the DPVM should not be used when $d/\Delta x > 1.0$. When the expanding

TABLE IV. Comparison between different methods, such as the DPVM, the optimized big particle method, and the diffusion-based method. The number in computation cost means the multiple of the CPU time spent on solving the fluid phase governing equations.

Methods	DPVM	Optimized big particle method	Diffusion-based method
Application range	$\Delta x/d > 1.0$	$\Delta x/d > 0.5$	$\Delta x/d > 0.5$
Computation cost	4.91	4.37	4.82
Model implementation	Difficult	Easy	Easy
Polydisperse effects	Considered	Considered	Not considered

factor is small, the particle volume fraction obtained from the simplified two-grid method and the optimized big particle method shows oscillation, which leads to large errors in calculation results. The expanding factor should be set as four or five in the big particle method based on comparison. The diffusion-based method does not consider the polydisperse effects because it does not consider the variation in particle sizes and sets the same diffusion coefficient for the small and big particles. It results in an even smoothing of exchanged information across all particles with different sizes. However, the optimized big particle method does consider the polydisperse effects because the expansion size of the particles is related to the particle size. As a result, the optimized big particle gives more consistent results with the experimental phenomena than the diffusion-based method in the binary fluidized bed discussed in this paper. In addition, the simplified two-grid method and the optimized big particle method are both easy for code implementation and require less computation costs than the diffusion-based method.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant Nos. 12302516 and 12132018) and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA22000000).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Yan Zhang: Data curation (equal); Funding acquisition (equal); Methodology (equal); Software (equal); Validation (equal); Writing – original draft (equal). **Wan-Long Ren:** Investigation (equal); Validation (equal); Writing – review & editing (equal). **Peng Li:** Methodology (equal); Validation (equal); Writing – review & editing (equal). **Xuhui Zhang:** Data curation (equal); Funding acquisition (equal); Software (equal); Writing – review & editing (equal). **Xiaobing Lu:** Methodology (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: VOLUMETRIC TWO-PHASE INTERACTION FORCE

The mass and momentum conservation equations of the fluid phase are written as[38]

$$\frac{(\partial \rho_f \alpha_f)}{\partial t} + \nabla \cdot (\rho_f \alpha_f \mathbf{u}) = 0, \tag{A1}$$

$$\frac{\partial (\rho_f \alpha_f \mathbf{u})}{\partial t} + \nabla \cdot (\rho_f \alpha_f \mathbf{u}\mathbf{u}) = -\alpha_f \nabla p - \mathbf{F}_{\mathrm{pf}} + \nabla \cdot (\alpha_f \rho_f \boldsymbol{\tau}) + \rho_f \alpha_f \mathbf{g}, \tag{A2}$$

where $\mathbf{F}_{\mathrm{pf}}$ is the volumetric two-phase interaction force, which is calculated by

$$\mathbf{F}_{\mathrm{pf}} = \frac{1}{V_{\mathrm{c}}} \sum_{i=1}^{k} (\mathbf{f}_{\mathrm{d},i} + \mathbf{f}_{\mathrm{lift},i} + \mathbf{f}_{\mathrm{vm},i}), \tag{A3}$$

where $\mathbf{f}_{\mathrm{d},i}$ is the drag force, $\mathbf{f}_{\mathrm{lift},i}$ is the lift force, $\mathbf{f}_{\mathrm{vm},i}$ is the virtual mass force, and k is the number of particles in a CFD cell. The drag force $\mathbf{f}_{\mathrm{d}}$ is calculated as follows:[16]

$$\mathbf{f}_{\mathrm{d}} = \frac{\pi}{8} C_{\mathrm{d}} \rho_f d^2 \alpha_f^{2-\chi} |\mathbf{u}_r| \mathbf{u}_r, \tag{A4}$$

$$C_{\mathrm{d}} = \left(0.63 + \frac{4.8}{\sqrt{Re}}\right)^2, \tag{A5}$$

$$\chi = 3.7 - 0.65 \exp\left(-\frac{(1.5 - \log Re)^2}{2}\right), \tag{A6}$$

$$Re = \frac{|\mathbf{u}_r| d}{\nu_f}, \tag{A7}$$

where C_{d} is the drag coefficient, χ is an empirical constant, $\mathbf{u}_r = \mathbf{u} - \mathbf{v}$ is the relative velocity between the fluid and particle, $\mathbf{v}$ is the particle velocity, d is the particle diameter, and ν_f is the viscosity of the fluid phase. The virtual mass force is written as

$$\mathbf{f}_{\mathrm{vm}} = -\frac{\rho_f V_{\mathrm{p}}}{2} \left(\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} - \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t}\right), \tag{A8}$$

where $\mathrm{D}/\mathrm{D}t$ is the material derivative. The lift force is calculated by

$$\mathbf{f}_{\mathrm{lift}} = \frac{\pi d^2}{4} \rho_f C_L |\mathbf{u}_r| \left((\mathbf{u}_r) \times \frac{\nabla \times \mathbf{u}}{|\nabla \times \mathbf{u}|}\right), \tag{A9}$$

where C_L is the lift coefficient and is calculated based on the model proposed by Kurose and Komori.[18]

APPENDIX B: DERIVATION OF THE FLUID PHASE EQUATIONS

Extracting the density item from Eqs. (A1) and (A2), the mass conservation equations of the fluid phase can be rewritten as

$$\begin{aligned} &\alpha_f \frac{\partial \rho_f}{\partial t} + \rho_f \frac{\partial \alpha_f}{\partial t} + \rho_f \nabla \cdot (\alpha_f \mathbf{u}) + \alpha_f \mathbf{u} \cdot \nabla \rho_f = 0, \\ &\rho_f \left(\frac{\partial \alpha_f}{\partial t} + \nabla \cdot (\alpha_f \mathbf{u})\right) = -\alpha_f \left(\frac{\partial \rho_f}{\partial t} + \mathbf{u} \cdot \nabla \rho_f\right) = -\alpha_f \frac{D\rho_f}{Dt}. \end{aligned} \tag{B1}$$

The left-hand side of Eq. (A2) is rewritten as

$$\begin{aligned} &\frac{\partial (\rho_f \alpha_f \mathbf{u})}{\partial t} + \nabla \cdot (\rho_f \alpha_f \mathbf{u}\mathbf{u}) \\ &= \rho_f \left(\frac{\partial (\alpha_f \mathbf{u})}{\partial t} + \nabla \cdot (\alpha_f \mathbf{u}\mathbf{u})\right) + \alpha_f \mathbf{u} \left(\frac{\partial \rho_f}{\partial t} + \mathbf{u} \cdot \nabla \rho_f\right) \\ &= \rho_f \left(\frac{\partial (\alpha_f \mathbf{u})}{\partial t} + \nabla \cdot (\alpha_f \mathbf{u}\mathbf{u})\right) + \alpha_f \mathbf{u} \frac{D\rho_f}{Dt}. \end{aligned} \tag{B2}$$

Substituting Eq. (B1) into Eq. (B2), we obtain

$$\frac{\partial(\rho_f\alpha_f\mathbf{u})}{\partial t}+\nabla\cdot(\rho_f\alpha_f\mathbf{uu}) = \rho_f\left(\frac{\partial(\alpha_f\mathbf{u})}{\partial t}+\nabla\cdot(\alpha_f\mathbf{uu})\right)-\rho_f\mathbf{u}\left(\frac{\partial\alpha_f}{\partial t}+\nabla\cdot(\alpha_f\mathbf{u})\right). \tag{B3}$$

Then, Eq. (A2) can be rewritten as

$$\frac{\partial}{\partial t}(\alpha_f\mathbf{u})+\nabla\cdot(\alpha_f\mathbf{uu})-\mathbf{u}\left(\frac{\partial\alpha_f}{\partial t}+\nabla\cdot(\alpha_f\mathbf{u})\right)-\nabla\cdot(\alpha_f\boldsymbol{\tau}) = -\alpha_f\nabla\frac{p}{\rho_f}-\frac{\mathbf{F}_{\mathrm{pf}}}{\rho_f}+\alpha_f\mathbf{g}. \tag{B4}$$

REFERENCES

[1]M. Maxey, "Simulation methods for particulate flows and concentrated suspensions," Annu. Rev. Fluid Mech. **49**, 171–193 (2017).
[2]D. J. Jerolmack and K. E. Daniels, "Viewing earth's surface as a soft-matter landscape," Nat. Rev. Phys. **1**, 716–730 (2019).
[3]J. Wang, "Continuum theory for dense gas-solid flow: A state-of-the-art review," Chem. Eng. Sci. **215**, 115428 (2020).
[4]Q. Guo, A. Bordbar, L. Ma, Y. Yu, S. Xu, C. M. Boyce, and M. Ye, "A CFD-DEM study of the solid-like and fluid-like states in the homogeneous fluidization regime of Geldart A particles," AIChE J. **68**, e17420 (2022).
[5]M. Ishii, "Two-fluid model for two-phase flow," Multiphase Sci. Technol. **5**, 1 (1990).
[6]Z. Malikov and M. Madaliev, "Numerical simulation of two-phase flow in a centrifugal separator," Fluid Dyn. **55**, 1012–1028 (2020).
[7]C.-H. Lee, "Multi-phase flow modeling of submarine landslides: Transformation from hyperconcentrated flows into turbidity currents," Adv. Water Resour. **131**, 103383 (2019).
[8]X. Cao, K. Yang, H. Wang, and J. Bian, "Modelling of hydrate dissociation in multiphase flow considering particle behaviors, mass and heat transfer," Fuel **306**, 121655 (2021).
[9]E. M. Fontalvo, R. L. C. Branco, J. N. Carneiro, and A. O. Nieckele, "Assessment of closure relations on the numerical predictions of vertical annular flows with the two-fluid model," Int. J. Multiphase Flow **126**, 103243 (2020).
[10]F. D. Cúnez and E. M. Franklin, "Crystallization and jamming in narrow fluidized beds," Phys. Fluids **32**, 083303 (2020).
[11]T. Zhang, X. Zeng, J. Guo, F. Zeng, and M. Li, "Numerical simulation on oil-water-particle flows in complex fractures of fractured-vuggy carbonate reservoirs," J. Pet. Sci. Eng. **208**, 109413 (2022).
[12]Z. Zhu, L. Xue, J. Sun, B. Cui, X. Su, and X. Xi, "Particle movement and hydraulic impact in dense two-phase solid–liquid flow inside a water–iron sand jet," Phys. Fluids **35**, 053313 (2023).
[13]D. P. Willen, A. J. Sierakowski, G. Zhou, and A. Prosperetti, "Continuity waves in resolved-particle simulations of fluidized beds," Phys. Rev. Fluids **2**, 114305 (2017).
[14]Y. Yao, C. S. Criddle, and O. B. Fringer, "Competing flow and collision effects in a monodispersed liquid–solid fluidized bed at a moderate Archimedes number," J. Fluid Mech. **927**, A28 (2021).
[15]Z. Wang and M. Liu, "On the determination of grid size/smoothing distance in un-/semi-resolved CFD-DEM simulation of particulate flows," Powder Technol. **394**, 73–82 (2021).
[16]R. Di Felice, "The voidage function for fluid-particle interaction systems," Int. J. Multiphase Flow **20**, 153–159 (1994).
[17]D. Gidaspow, *Multiphase Flow and Fluidization: Continuum and Kinetic Theory Descriptions* (Academic Press, San Diego, 1994).
[18]R. Kurose and S. Komori, "Drag and lift forces on a rotating sphere in a linear shear flow," J. Fluid Mech. **384**, 183–206 (1999).
[19]C. Wan, S. Xiao, D. Zhou, H. Zhu, Y. Bao, K. Kakanda, and Z. Han, "Numerical simulation on transport behavior of gradated coarse particles in deep-sea vertical pipe transportation," Phys. Fluids **35**, 043328 (2023).
[20]N. Deen, M. Van Sint Annaland, M. Van der Hoef, and J. Kuipers, "Review of discrete particle modeling of fluidized beds," Chem. Eng. Sci. **62**, 28–44 (2007).
[21]C. Wu, A. Berrouk, and K. Nandakumar, "Three-dimensional discrete particle model for gas–solid fluidized beds on unstructured mesh," Chem. Eng. J. **152**, 514–529 (2009).
[22]C. Wu, J. Zhan, Y. Li, K. Lam, and A. Berrouk, "Accurate void fraction calculation for three-dimensional discrete particle model on unstructured mesh," Chem. Eng. Sci. **64**, 1260–1266 (2009).
[23]S. Deb and D. K. Tafti, "A novel two-grid formulation for fluid–particle systems using the discrete element method," Powder Technol. **246**, 601–616 (2013).
[24]B. J. Glasser and I. Goldhirsch, "Scale dependence, correlations, and fluctuations of stresses in rapid granular flows," Phys. Fluids **13**, 407–420 (2001).
[25]Y. Zhang, Y. Zhao, Z. Gao, C. Duan, J. Xu, L. Lu, J. Wang, and W. Ge, "Experimental and Eulerian-Lagrangian-Lagrangian study of binary gas-solid flow containing particles of significantly different sizes," Renewable Energy **136**, 193–201 (2019).
[26]R. Sun and H. Xiao, "Diffusion-based coarse graining in hybrid continuum–discrete solvers: Applications in CFD–DEM," Int. J. Multiphase Flow **72**, 233–247 (2015).
[27]R. Sun and H. Xiao, "Diffusion-based coarse graining in hybrid continuum–discrete solvers: Theoretical formulation and a priori tests," Int. J. Multiphase Flow **77**, 142–157 (2015).
[28]Y. Zhang, X. Lu, and X. Zhang, "An optimized Eulerian-Lagrangian method for two-phase flow with coarse particles: Implementation in open-source field operation and manipulation, verification, and validation," Phys. Fluids **33**, 113307 (2021).
[29]C. Kloss, C. Goniva, A. Hager, S. Amberger, and S. Pirker, "Models, algorithms and validation for opensource DEM and CFD–DEM," Prog. Comput. Fluid Dyn. **12**, 140–152 (2012).
[30]A. Volk, U. Ghia, and C. Stoltz, "Effect of grid type and refinement method on CFD-DEM solution trend with grid size," Powder Technol. **311**, 137–146 (2017).
[31]J. Zhang, T. Li, H. Strom, and T. Lovas, "Grid-independent Eulerian-Lagrangian approaches for simulations of solid fuel particle combustion," Chem. Eng. J. **387**, 123964 (2020).
[32]Z. Wang, Y. Teng, and M. Liu, "A semi-resolved CFD–DEM approach for particulate flows with kernel based approximation and Hilbert curve based searching strategy," J. Comput. Phys. **384**, 151–169 (2019).
[33]J. Zhang, T. Li, H. Ström, B. Wang, and T. Lùvås, "A novel coupling method for unresolved CFD-DEM modeling," Int. J. Heat Mass Transfer **203**, 123817 (2023).
[34]H. Jasak, "OpenFOAM: Open source CFD in research and industry," Int. J. Nav. Archit. Ocean Eng. **1**, 89–94 (2009).
[35]Z. Peng, E. Doroodchi, C. Luo, and B. Moghtaderi, "Influence of void fraction calculation on fidelity of CFD-DEM simulation of gas-solid bubbling fluidized beds," AIChE J. **60**, 2000–2018 (2014).
[36]S. Khorasanizade and J. Sousa, "Improving linked-lists using tree search algorithms for neighbor finding in variable-resolution smoothed particle hydrodynamics," Comput. Phys. Commun. **26**, 57–86 (2019).
[37]W. Ren, X. Zhang, Y. Zhang, P. Li, and X. Lu, "Investigation of particle size impact on dense particulate flows in a vertical pipe," Phys. Fluids **35**, 073302 (2023).
[38]Z. Y. Zhou, S. B. Kuang, K. W. Chu, and A. B. Yu, "Discrete particle simulation of particle–fluid flow: Model formulations and their applicability," J. Fluid Mech. **661**, 482–510 (2010).
[39]Z. Zhang, J. Wang, R. Huang, R. Qiu, X. Chu, S. Ye, Y. Wang, and Q. Liu, "Data-driven turbulence model for unsteady cavitating flow," Phys. Fluids **35**, 015134 (2023).
[40]J. Capecelatro and O. Desjardins, "An Euler–Lagrange strategy for simulating particle-laden flows," J. Comput. Phys. **238**, 1–31 (2013).
[41]Y. Tsuji, T. Kawaguchi, and T. Tanaka, "Discrete particle simulation of two-dimensional fluidized bed," Powder Technol. **77**, 79–87 (1993).
[42]D. Li and H. Christian, "Simulation of bubbly flows with special numerical treatments of the semi-conservative and fully conservative two-fluid model," Chem. Eng. Sci. **174**, 25–39 (2017).
[43]C. M. Rhie and W. L. Chow, "Numerical study of the turbulent flow past an airfoil with trailing edge separation," AIAA J. **21**, 1525–1532 (1983).
[44]S. Zhang, X. Zhao, and S. Bayyuk, "Generalized formulations for the Rhie–Chow interpolation," J. Comput. Phys. **258**, 880–914 (2014).

[45]H. Jasak, "Error analysis and estimation for the finite volume method with applications to fluid flows," Ph.D. thesis (Imperial College London, 1996).
[46]D. Li, D. Marchisio, C. Hasse, and D. Lucas, "Comparison of Eulerian QBMM and classical Eulerian–Eulerian method for the simulation of polydisperse bubbly flows," AIChE J. **65**, e16732 (2019).
[47]D. Li, D. Marchisio, C. Hasse, and D. Lucas, "twoWayGPBEFoam: An open-source Eulerian QBMM solver for monokinetic bubbly flows," Comput. Phys. Commun. **250**, 107036 (2020).
[48]E. A. Schnorr Filho, N. C. Lima, and E. M. Franklin, "Resolved CFD-DEM simulations of the hydraulic conveying of coarse grains through a very-narrow elbow," Powder Technol. **395**, 811–821 (2022).
[49]Y. Zhang, W. Ren, P. Li, X. Zhang, and X. Lu, "Flow regimes and characteristics of dense particulate flows with coarse particles in inclined pipe," Powder Technol. **428**, 118859 (2023).
[50]C. Müller, D. Holland, A. Sederman, S. Scott, J. Dennis, and L. Gladden, "Granular temperature: Comparison of magnetic resonance measurements with discrete element model simulations," Powder Technol. **184**, 241–253 (2008).
[51]C. R. Müller, S. A. Scott, D. J. Holland, B. C. Clarke, A. J. Sederman, J. S. Dennis, and L. F. Gladden, "Validation of a discrete element model using magnetic resonance measurements," Particuology **7**, 297–306 (2009).
[52]Y. Zhang, Y. Liu, W. Ren, P. Li, X. Zhang, and X. Lu, "Kinematic waves and collision effects in dense fluid–particle flow during hydraulic conveying," Int. J. Multiphase Flow **170**, 104643 (2024).
[53]R. Sun and H. Xiao, "SediFoam: A general-purpose, open-source CFD–DEM solver for particle-laden flow with emphasis on sediment transport," Comput. Geosci. **89**, 207–219 (2016).
[54]R. Sun and H. Xiao, "CFD–DEM simulations of current-induced dune formation and morphological evolution," Adv. Water Resour. **92**, 228–239 (2016).
[55]J. Zeng, P. Tang, H. Li, and D. Zhang, "Simulating particle settling in inclined narrow channels with the unresolved CFD-DEM method," Phys. Rev. Fluids **6**, 034302 (2021).
[56]M. S. Khan, S. Mitra, S. Ghatage, Z. Peng, E. Doroodchi, B. Moghtaderi, J. B. Joshi, and G. M. Evans, "Pressure drop and voidage measurement in solid-liquid fluidized bed: Experimental, mathematical and computational study," in *Proceedings of the Chemeca 2016: Chemical Engineering-Regeneration, Recovery and Reinvention* (Engineers Australia, 2016), pp. 1019–1030.
[57]J. Richardson and W. Zaki, "The sedimentation of a suspension of uniform spheres under conditions of viscous flow," Chem. Eng. Sci. **3**, 65–73 (1954).
[58]U. Ganguly, "On the prediction of terminal settling velocity of solids in liquid-solid systems," Int. J. Miner. Process. **29**, 235–247 (1990).
[59]J. Garside and M. R. Al-Dibouni, "Velocity-voidage relationships for fluidization and sedimentation in solid-liquid systems," Ind. Eng. Chem. Process Des. Dev. **16**, 206–214 (1977).
[60]C.-C. Huang, J. A. van Oijen, N. G. Deen, and Y. Tang, "A particle-size dependent smoothing scheme for polydisperse Euler-Lagrange simulations," Chem. Eng. Sci. **277**, 118765 (2023).