

## Research Article

# Improving the Accuracy of Fictitious Domain Method Using Indicator Function from Volume Intersection

Guoliang Chai <sup>1</sup>, Junwei Su <sup>1</sup>, Le Wang,<sup>2</sup> Chunlei Yu,<sup>3</sup> Yigen Zhang,<sup>3</sup> and Zhaolin Gu<sup>1</sup>

<sup>1</sup>School of Human Settlements and Civil Engineering, Xi'an Jiaotong University, Xi'an 710049, China

<sup>2</sup>School of Mechanical Engineering, Xi'an Shiyou University, Xi'an 710056, China

<sup>3</sup>Research Institute of Exploration and Development, Shengli Oilfield Company, Sinopec Group, Dongying 257015, China

Correspondence should be addressed to Junwei Su; [sujunwei@mail.xjtu.edu.cn](mailto:sujunwei@mail.xjtu.edu.cn)

Received 22 February 2019; Revised 13 May 2019; Accepted 22 May 2019; Published 26 June 2019

Academic Editor: Ivan Giorgio

Copyright © 2019 Guoliang Chai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fictitious domain method (FDM) is a commonly accepted direct numerical simulation technique for moving boundary problems. Indicator function used to distinguish the solid zone and the fluid zone is an essential part concerning the whole prediction accuracy of FDM. In this work, a new indicator function through volume intersection is developed for FDM. In this method, the arbitrarily polyhedral cells across the interface between fluid and solid are located and subdivided into tetrahedrons. The fraction of the solid volume in each cell is accurately computed to achieve high precision of integration calculation in the particle domain, improving the accuracy of the whole method. The quadrature over the solid domain shows that the newly developed indicator function can provide results with high accuracy for variable integration in both stationary and moving boundary problems. Several numerical tests, including flow around a circular cylinder, a single sphere in a creeping shear flow, settlement of a circular particle in a closed container, and in-line oscillation of a circular cylinder, have been performed. The results show good accuracy and feasibility in dealing with the stationary boundary problem as well as the moving boundary problem. This method is accurate and conservative, which can be a feasible tool for studying problems with moving boundaries.

## 1. Introduction

Moving boundary problems appear widely in nature (freely falling leaves) and engineering (particle flow, rotating blades of rotating machinery). In such problems, the boundary movement disturbs the flow of the surrounding fluid, resulting in complicated flow behaviour, which affects the dynamics of the moving object in turn and further changes its trajectory. There is a complicated nonlinear coupling relationship between flow behaviour and its trajectory. It is essential to track the influence of boundary movement on the flow and predict the complex behaviour of the moving boundary system accurately by the feedback of the flow field to the moving object.

With the development of the computer, computational fluid dynamics (CFD) has become an important method to study the moving boundary problems. At present, the numerical simulation methods for moving boundary problems can be divided into the point-source method [1–3] and direct

numerical simulation method [4–15]. In the point-source method, fluid-solid interactions are evaluated through point source models (drag model, lift model, etc.) and the solid (or particle) is much smaller than the mesh size. This situation is beyond the scope of this work. Direct numerical simulation techniques include arbitrary Lagrange-Euler (ALE) [4–6] and fixed mesh method [7–15]. For ALE, body-fitted mesh is usually applied, and the force and moment of fluid acting on the solid are integrated by the force on the boundary; then, the solid is moved and rotated by tracking the dynamic process and kinematics equation of the solid; finally, the fluid mesh is adjusted to calculate the flow for the next moment. This method can track the flow of fluid precisely. In this method, the fluid mesh needs to be moved when the solid is moving. If the solid domain changes greatly (e.g., the problem of rotating machinery), the mesh will greatly deform, resulting in a great decrease of mesh quality without change of mesh topology, affecting the prediction accuracy. If the mesh topology is changed, more complicated

mesh reconstruction will be needed. This method applies only if the solid domain barely changes. For the fixed mesh method, the fluid mesh does not change during the solution procedure, and the behaviour of fluid is changed in the solid domain or solid boundary-covered mesh to indicate the effects of the solid on the flow. The immersed boundary method [7, 8] and fictitious domain method [9–15] belong to such methods. These methods avoid complicated mesh operations, and thus it has become an important method of dealing with moving boundary problems in recent years [16].

Fictitious domain method is an important method for solving the moving boundary problems. This method divides the solution domain into fluid covering domain and solid covering domain. In the solid covering domain, a pseudo body force is exerted to make the fluid behaviour become consistent with the behaviour of the solid. This method was first proposed by Glowinski et al. [9]. They introduced the body force as a Lagrange multiplier to the momentum equation and solved it by iteration. Because of the saddle point problem, the computational load is increased. In order to overcome this disadvantage, Sharma and Patankar [11] proposed a non-Lagrangian multiplier/fictitious domain method, which simplifies the computation by calculating the body force explicitly, thereby significantly reducing the computational load. However, this method defines the body force at some Lagrangian points on the solid domain, while the particle moves in the Eulerian background mesh. Thus, the scheme of the body force mapping between the Eulerian mesh and the Lagrangian point determines the numerical accuracy. The nonsmoothness of the field obtained by the mapping incurs numerical instability. To overcome this problem, Gallier et al. [14] proposed a fictitious domain method in a fully Eulerian framework. The body force is defined in the centre of the mesh cell, rather than Lagrangian points. All the solution procedures are performed on the Eulerian mesh, and the information exchange between the Eulerian framework and the Lagrangian frame is avoided. However, the above process is achieved by defining a particle indicator function, which is 0 outside the particle and 1 inside the particle and smoothly transits from 0 to 1 within a grid near the particle boundary. The smooth transition process is specified by a Heaviside function, rather than the volume ratio of the particle domain to the arbitrarily polyhedral mesh cells, so the integral calculated by this particle indicator function does not possess the feature of conservation.

Based on the fully Eulerian fictitious domain method of Gallier et al., a fictitious domain method based on the particle indicator function using intersection volume is proposed in this paper. This method marks the Eulerian arbitrarily polyhedral mesh cells at the interface between fluid and particles, firstly. The mesh cells are divided into a serial of tetrahedrons by a cell decomposing technique, and thus, the intersection volume between the particle and the mesh cell can be evaluated accurately. Adopting the interaction volume as the indicator function can guarantee the accuracy and conservation of the integral of the particle domain, thereby enhancing the overall numerical accuracy.

## 2. Fictitious Domain Method

In this paper, Gallier et al.'s fictitious domain method [14] is employed as an example to test the volume-intersection based indicator function. In order to make the method adapt to arbitrarily polyhedral mesh, the collocated finite volume method is employed in this work rather than the finite-difference method adopted in original work.

*2.1. Governing Equations.* The computational domain  $D$  contains fluid domain  $D_f$  and particle domain  $D_p$ , i.e.,  $D = D_f \cup D_p$ . The particle domain contains  $N$  rigid particles  $P_i$  with the density of  $\rho_p$ , and the interface between particle and fluid is  $\partial P$ . The fluid is incompressible viscous Newtonian fluid with the density of  $\rho_f$ , the dynamic viscous coefficient of  $\mu$ , and the kinematic viscous coefficient of  $\nu = \mu/\rho_f$ . The governing equation of the fictitious domain model is expressed as follows

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \nabla \cdot \mathbf{u} = -\frac{\nabla p}{\rho_f} + \nu \nabla^2 \mathbf{u} + \boldsymbol{\lambda} \quad \text{in } D \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } D \quad (2)$$

$$\mathbf{u} = \mathbf{U} + \boldsymbol{\Omega} \times \mathbf{r} \quad \text{in } D_p \quad (3)$$

$$\rho_f \int_P \boldsymbol{\lambda} d\mathbf{x} = \frac{\rho_f - \rho_p}{\rho_p} M \left( \frac{d\mathbf{U}}{dt} - \mathbf{g} \right) + \mathbf{F}^{int} \quad (4)$$

$$\rho_f \int_P \mathbf{r} \times \boldsymbol{\lambda} d\mathbf{x} = \frac{\rho_f - \rho_p}{\rho_p} \left( \mathbf{J} \cdot \frac{d\boldsymbol{\Omega}}{dt} - \boldsymbol{\Omega} \times (\mathbf{J} \cdot \boldsymbol{\Omega}) \right) + \mathbf{T}^{int} \quad (5)$$

where  $\mathbf{u}$  and  $p$  are the velocity and pressures of fluid, respectively;  $\boldsymbol{\lambda}$  is a pseudo body force, which is nonzero in the particle domain and zero elsewhere, and therefore the Navier-Stokes equation is also valid in the particle domain;  $M$ ,  $\mathbf{J}$ ,  $\mathbf{U}$ , and  $\boldsymbol{\Omega}$  are the mass, moment of inertia, translational velocity, and rotational angular velocity of a single particle, respectively;  $\mathbf{r}$  is the position vector at the mass centre of a particle;  $\mathbf{g}$  is gravitational acceleration;  $\mathbf{F}^{int}$  and  $\mathbf{T}^{int}$  are the forces and moments acting on the particle except for the fluid force.

In the above governing equations, (1) and (2) are the momentum equation and continuity equation satisfied in the whole solution domain, respectively. The body force,  $\boldsymbol{\lambda}$ , is enforced in the solid domain to make momentum equation valid in the whole solution domain. Equation (3) enforces rigid body constraint in the particle domain. Equations (4) and (5) are particle dynamic equations.

*2.2. Numerical Procedure.* Two subproblems can be divided based on the above equation descriptions, i.e., the fluid subproblems of (1) and (2) and the particle subproblems of (3)-(5).

*2.2.1. Fluid Subproblem.* Equation (1) is the equation describing the fluid dynamics, which Gallier et al. [14] used the

finite-difference method to discretize. In this paper, the finite volume method is employed. The semidiscrete form of (1) can be expressed as

$$a_p \mathbf{u} = \mathbf{A}_H - \frac{\nabla p}{\rho_f} \quad (6)$$

where  $a_p$  is the diagonal value of the coefficient matrix obtained by discretization of (1), and  $\mathbf{A}_H$  is expressed as

$$\mathbf{A}_H = \sum_N a_N \mathbf{u} + \mathbf{b} \quad (7)$$

where  $a_N$  is the implicit contribution coefficient of the neighbour cells to the current cell during discretization of (1) and  $\mathbf{b}$  is the explicit discretization contribution except pressure. Since  $\lambda$  is unknown at this time, the explicit discretization is used here.

Both sides of (6) are divided by  $a_p$  yielding

$$\mathbf{u} = \frac{\mathbf{A}_H}{a_p} - \frac{\nabla p}{\rho_f a_p} \quad (8)$$

The velocity field obtained by (8) should satisfy the continuity equation. Substituting (8) into (2) yields

$$\nabla \cdot \left( \frac{1}{\rho_f a_p} \nabla p \right) = \nabla \cdot \left( \frac{\mathbf{A}_H}{a_p} \right) \quad (9)$$

Equation (9) is a pressure equation derived by PISO algorithm [19]. A new pressure can be obtained by solving the equation. Equation (8) is used to update the velocity. In this paper, the collocated finite volume method is used for discretization. While the velocity on the mesh cell centre is updated, it is necessary to update the flow rate at the cell face using

$$\varphi_f = \left( \frac{\mathbf{A}_H}{a_p} \right)_f \cdot \mathbf{S}_f - \left( \frac{\nabla p}{\rho_f a_p} \right)_f \cdot \mathbf{S}_f \quad (10)$$

where  $(\mathbf{m})_f$  is the interpolation of value of the body centre,  $\mathbf{m}$ , on the face centre,  $\mathbf{S}_f$  is the area vector of the cell face, and  $\varphi_f$  is the volume flow rate on the cell face.

**2.2.2. Particle Subproblem.** For the particle subproblem, the following equations are solved first

$$\begin{aligned} M \mathbf{U}^{n+1} &= \frac{\rho_P - \rho_f}{\rho_P} M (\mathbf{U}^n + \mathbf{g} \Delta t) \\ &+ \rho_f \int_P (\mathbf{u}^{n+1} - \lambda^n \Delta t) d\mathbf{x} + \mathbf{F}^{int,n} \Delta t \end{aligned} \quad (11)$$

$$\begin{aligned} \mathbf{J} \cdot \boldsymbol{\Omega}^{n+1} &= \frac{\rho_P - \rho_f}{\rho_P} (\mathbf{J} \cdot \boldsymbol{\Omega}^n - \boldsymbol{\Omega}^n \times (\mathbf{J} \cdot \boldsymbol{\Omega}^n) \Delta t) \\ &+ \rho_f \int_P \mathbf{r} \times (\mathbf{u}^{n+1} - \lambda^n \Delta t) d\mathbf{x} \\ &+ \mathbf{T}^{int,n} \Delta t \end{aligned} \quad (12)$$

to obtain the translational velocity,  $\mathbf{U}^{n+1}$ , and rotational angular velocity,  $\boldsymbol{\Omega}^{n+1}$ , at the new time step. Then, use the equation below

$$\tilde{\mathbf{u}}^{n+1} = \mathbf{U}^{n+1} + \boldsymbol{\Omega}^{n+1} \times \mathbf{r} \quad \text{in } D_P \quad (13)$$

to update the fluid velocity in the particle covering domain and use this velocity to update the pseudo body force,  $\lambda^{n+1}$ .

$$\lambda^{n+1} = \lambda^n + \frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^{n+1}}{\Delta t} \quad (14)$$

Finally, the second-order Adams-Bashforth difference scheme is used to discretize the displacement equation and update the particle position.

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \frac{3}{2} \Delta t \mathbf{U}^{n+1} - \frac{1}{2} \Delta t \mathbf{U}^n \quad (15)$$

For nonspherical particle, it is necessary to rotate the particle by the angular velocity.

**2.2.3. Calculation of Integral of Particle Domain.** Equations (11) and (12) contain the integrals of physical quantities on the particle domain. As a DNS (direct numerical simulation) method, the faces of background mesh cells do not coincide with the particle surface during the solving procedure, so it is necessary to use an approximation method to deal with the interface area. Gallier et al. [14] used the traditional approximate Heaviside function to calculate the integral term. The integral of any physical quantity,  $\Phi$ , in the particle covering domain is approximated by

$$\int_P \Phi d\mathbf{x} = \int_{D_P} I(\mathbf{x}) \Phi d\mathbf{x} \quad (16)$$

where  $I(\mathbf{x})$  is the indicator function, which is expressed by the following expression

$$I(\mathbf{x}) = 1 - H(\psi(\mathbf{x})) \quad (17)$$

where  $\psi(\mathbf{x})$  is the level-set function and is expressed as

$$\psi(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_c| - r \quad (18)$$

where  $\mathbf{x}_c$  and  $r$  are the mass centre and radius of the particle, respectively;  $\psi(\mathbf{x})$  is a negative value inside the particle and a positive value outside the particle. The approximate Heaviside function can be expressed as

$$H(s) = \frac{1}{2} \left( 1 + \tanh \left( b \frac{s}{\Delta} \right) \right) \quad (19)$$

where  $\Delta$  is the grid spacing and the constant  $b=5$  guarantees the smooth width within a grid. Eventually, the particle indicator function  $I(\mathbf{x})$  is 1 inside the particle and 0 outside the particle and smoothly transits from 1 to 0 at the interface of the two phases.

The approximate Heaviside function guarantees a smooth transition within a grid around the interface and enhances the stability of the algorithm. However, it cannot guarantee

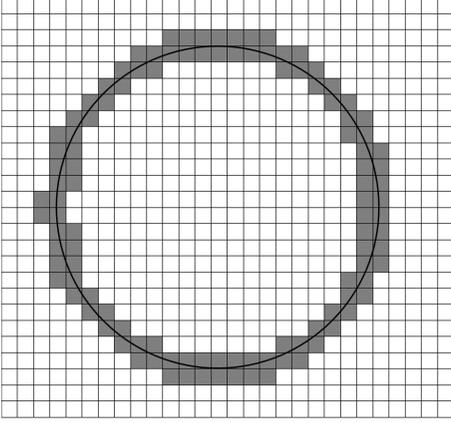


FIGURE 1: Mesh cells near the particle boundary.

the conservation. If the integrand is a constant, 1, the volume of the particle should be obtained by integration of (16). However, the use of approximate Heaviside function leads to a big difference between the calculated volume and actual volume of the particle, i.e., nonconservation of the volume. In the particle subproblem, (11)-(12) involve the integration of velocity and pseudo body force in the particle domain. The integration with approximate Heaviside function sacrifices the numerical accuracy.

### 3. Intersecting Volume-Based Particle Indicator Function

**3.1. Locating the Cells near the Solid Boundary.** Based on the level-set function from (18), the mesh cells of  $\psi(\mathbf{x}) > 0$  are set to  $I(\mathbf{x}) = 0$  first, and the rest are set to  $I(\mathbf{x}) = 1$ . The cells near the solid boundary are located using the level set functions  $\psi(\mathbf{x}_{owner})$  and  $\psi(\mathbf{x}_{neighbour})$ . The subscripts *owner* and *neighbour* represent the current cell and the neighbour cell, respectively. If  $\psi(\mathbf{x}_{owner}) \psi(\mathbf{x}_{neighbour}) < 0$ , two cells are located on both sides of the interface, respectively. Thus, they are marked as the cells near the solid interface. All the cells are checked to locate the mesh cells near the interfaces. This is illustrated in Figure 1, where the circular solid line is the actual boundary of the circled particle, and the shaded cells are the marked ones at the particle interface.

**3.2. Subdivision of Mesh Cells near the Solid Boundary.** In order to calculate the intersection volumes between the mesh cells and the particles accurately, the cells near the interface are decomposed into tetrahedrons with different levels. The arbitrarily polyhedral cell is taken as an example. The process is as follows.

Firstly, an arbitrary polyhedron is decomposed into several tetrahedrons.

Figure 2(a) is a polyhedral cell centred at  $X_i$ . The method given by Su et al. [20] is applied to decompose the polyhedron. In this method, we connect the centre to each vertex of the polyhedron and twelve pentagonal pyramids are obtained. One of the pyramids,  $\nu_0 \nu_1 \nu_2 \nu_3 \nu_4 X_i$ , is taken separately, as

shown in Figure 2(b), and  $X_f$  and  $S_f$  are the face centre and area vector of the pentagon  $\nu_0 \nu_1 \nu_2 \nu_3 \nu_4$ , respectively. Three tetrahedrons,  $\nu_0 \nu_1 \nu_2 X_i$ ,  $\nu_0 \nu_2 \nu_3 X_i$ , and  $\nu_0 \nu_3 \nu_4 X_i$ , can be obtained by connecting  $\nu_0$ ,  $\nu_2$ , and  $\nu_3$ . The above operation is repeated for the rest of the pyramids so that the polyhedron can be decomposed into 36 tetrahedrons as shown in Figure 2(c).

Secondly, a tetrahedron can be decomposed into eight small tetrahedrons further to improve the accuracy.

The tetrahedron shown in Figure 3(a) can be subdivided into eight subtetrahedrons. Four small tetrahedrons and one octahedron can be obtained by connecting any two of the midpoints of six edges of the tetrahedron along four surfaces as shown in Figure 3(b). The central octahedron given in Figure 3(c) can be subdivided into four tetrahedrons by connecting a pair of nonadjacent vertices, and the final tetrahedrons are shown in Figure 3(d). Finally, eight small tetrahedrons are obtained. According to this strategy, the tetrahedrons obtained can be decomposed into several smaller tetrahedrons further. The process of decomposing a polyhedron into several tetrahedrons is called Level 0 subdivision. The decomposed tetrahedron can be decomposed further if needed. Each subsequent decomposition process is called one higher level subdivision. The number of tetrahedrons after decomposition is eight times the number before the decomposition.

**3.3. Calculation of Particle Indicator Function  $I(\mathbf{x})$  of the Cells near the Solid Boundary.** After subdivision, the mesh cells near the solid boundary are divided into several tetrahedrons, which are in three states, outside the particle, inside the particle, and across the solid interface. For the last one, an algorithm similar to Rabbitz's algorithm [17] is used to determine the intersection points between the interface and the tetrahedron edges. Rabbitz determines the intersection points by planes intersection. In this paper, the level-set function obtained by (18) is used to determine the intersection points. For the level-set value of any two vertices,  $\nu_1$  and  $\nu_2$ , of a tetrahedron across the interface, if  $\psi(\mathbf{x}_{\nu_1}) \psi(\mathbf{x}_{\nu_2}) < 0$ , the edge containing  $\nu_1$  and  $\nu_2$  intersects with the interface, and the intersection point  $\mathbf{x}_i$  can be determined by computing  $\psi(\mathbf{x}_i) = 0$  on that edge.

Then a tetrahedron is divided into two parts by connecting any two of these intersection points along the tetrahedron surfaces (Figure 4(a), the shaded plane), and one shaded part is inside the particle (Figures 4(b) and 4(c)), which is either a tetrahedron or not. If not, it can be further decomposed into several tetrahedrons. And with all these tetrahedrons, the volume of the solid part is easy to obtain.

The solid volume in a cell near the solid boundary is the sum of the volumes,  $\sum V_p$ , of all geometries inside the particle (including the tetrahedrons inside and the shaded part of the tetrahedron across the interface that is inside). The solid volume fraction evaluated by  $\sum V_p / V_{cell}$  is adopted as the particle indicator function  $I(\mathbf{x})$  of the cell.

Notice that we assume the interface within the tetrahedron is flat (Figure 4(a)), which is not true. Numerical errors are inevitable. However, theoretically, a higher subdivision level leads to a smaller error. Thus, with sufficient levels

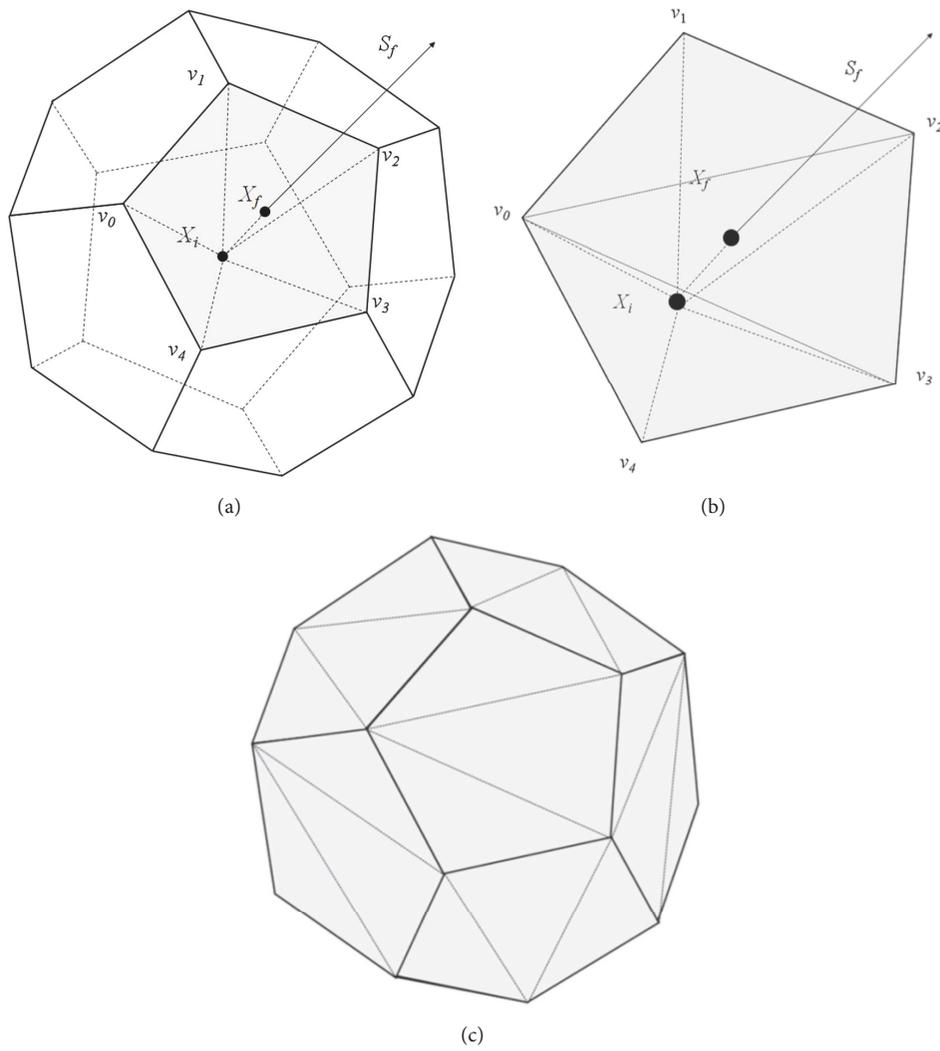


FIGURE 2: Decomposition of a polyhedral cell into tetrahedrons: (a) original polyhedron cut into pyramids; (b) a pyramid cut into tetrahedrons; (c) decomposition result.

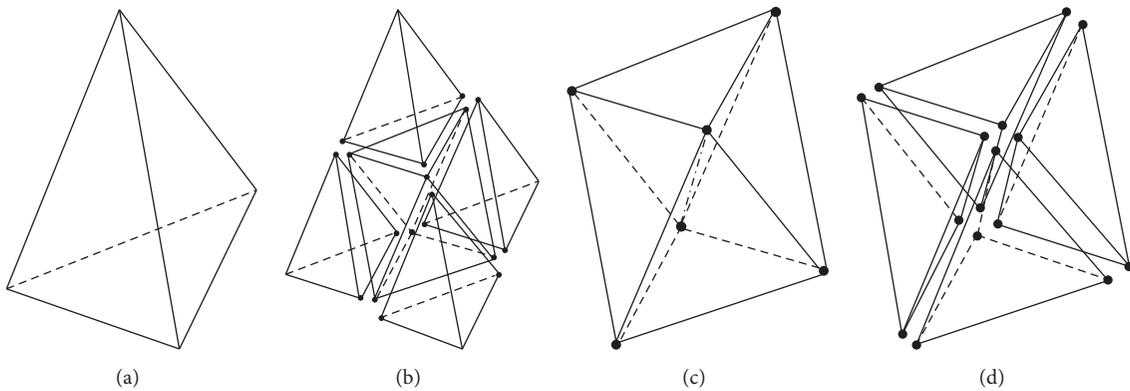


FIGURE 3: Subdivision of a tetrahedron: (a) original tetrahedron; (b) initial subdivision; (c) central octahedron; (d) subdivision of central octahedron [17].

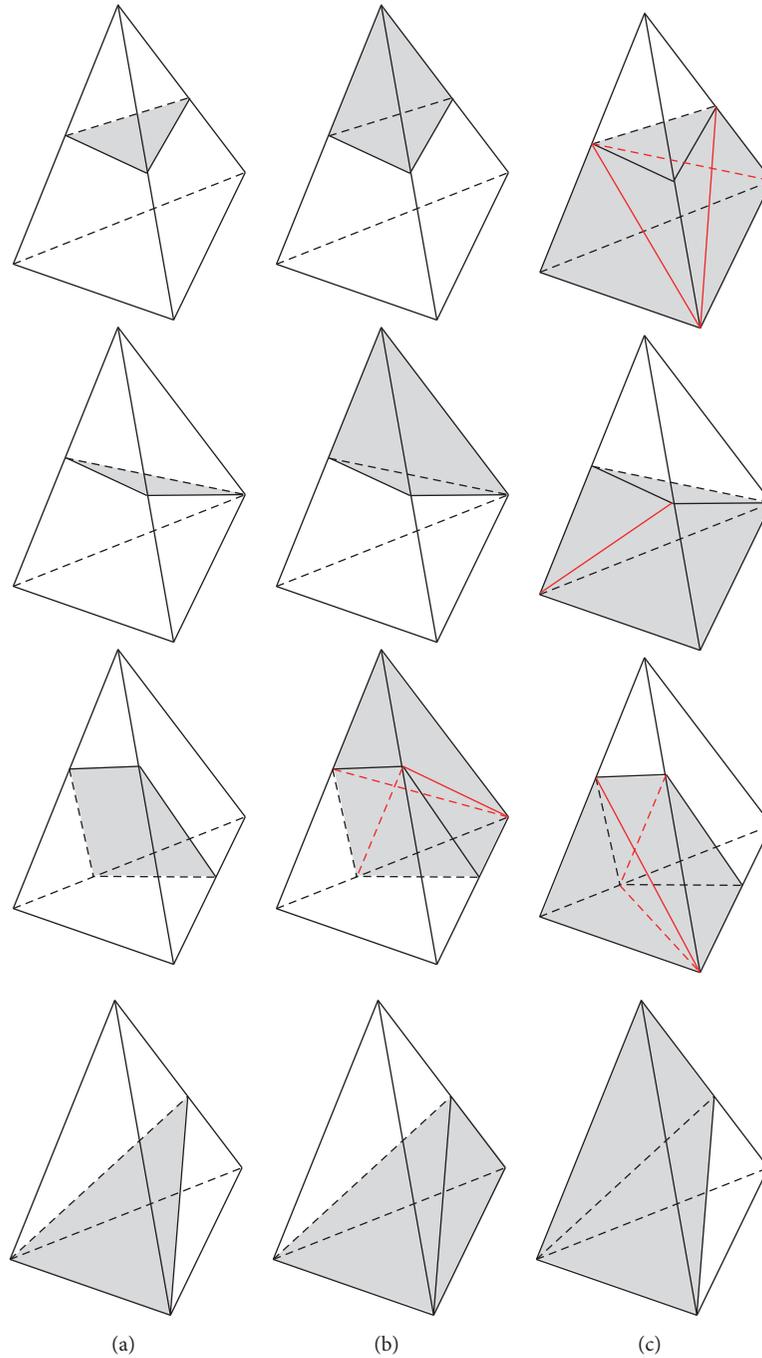


FIGURE 4: All the circumstances of (a) a tetrahedron intersected by a surface and resulting in (b) one or (c) another shaded geometry within the surface, which is already a tetrahedron or can be decomposed into several tetrahedrons by the red lines.

of the subdivision, the required numerical accuracy can be obtained.

## 4. Results and Discussion

**4.1. Particle Volume Evaluation.** In this example, the accuracy of the volume calculation of circular and spherical particles on 2D and 3D structured background mesh at different subdivision levels. The circular and spherical particles have

the diameter,  $d = 1$  m, the grid spacing is set to  $d/10$ ,  $d/20$ ,  $d/30$ , and  $d/40$ , respectively, and the subdivision level is set to 1, 2, 3, and 4, totally 16 cases. Tables 1 and 2 give the particle volumes  $\sum I(\mathbf{x})V_{cell}$  and relative errors by two indicator functions in 2D and 3D, respectively. The exact volumes for 2D circle and 3D volume are 0.785398 and 0.523599, respectively. It can be observed from the tables that, with a grid fining, our method produces much smaller relative errors than the approximate Heaviside function

TABLE 1: Comparison of results and errors of the particle volume computation in 2D.

Grid Spacing	Volume and Errors	Gallier [14]	Subdivision level			
			1	2	3	4
$d/10$	Volume	0.793864	0.785212	0.785347	0.785384	0.785394
	Error	1.08e-2	2.37e-4	6.50e-5	1.80e-5	5.02e-6
$d/20$	Volume	0.788934	0.785353	0.785386	0.785395	0.785397
	Error	4.50e-3	5.92e-5	1.56e-5	4.33e-6	1.21e-6
$d/30$	Volume	0.787999	0.785379	0.785393	0.785397	0.785398
	Error	3.31e-3	2.50e-5	6.90e-6	1.91e-6	5.31e-7
$d/40$	Volume	0.787515	0.785387	0.785395	0.785397	0.785398
	Error	2.70e-3	1.41e-5	3.88e-6	1.07e-6	2.96e-7

TABLE 2: Comparison of results and errors of the particle volume computation in 3D.

Grid spacing	Volume and Errors	Gallier [14]	Subdivision level			
			1	2	3	4
$d/10$	Volume	0.529842	0.523251	0.523503	0.523572	0.523591
	Error	1.19e-2	6.64e-4	1.83e-4	5.09e-5	1.42e-5
$d/20$	Volume	0.527607	0.523512	0.523575	0.523592	0.523597
	Error	7.66e-3	1.66e-4	4.60e-5	1.28e-5	3.56e-6
$d/30$	Volume	0.525825	0.52356	0.523588	0.523596	0.523598
	Error	4.25e-3	7.39e-5	2.04e-5	5.66e-6	1.58e-6
$d/40$	Volume	0.524104	0.523577	0.523593	0.523597	0.523598
	Error	9.65e-4	4.15e-5	1.15e-5	3.19e-6	8.94e-7

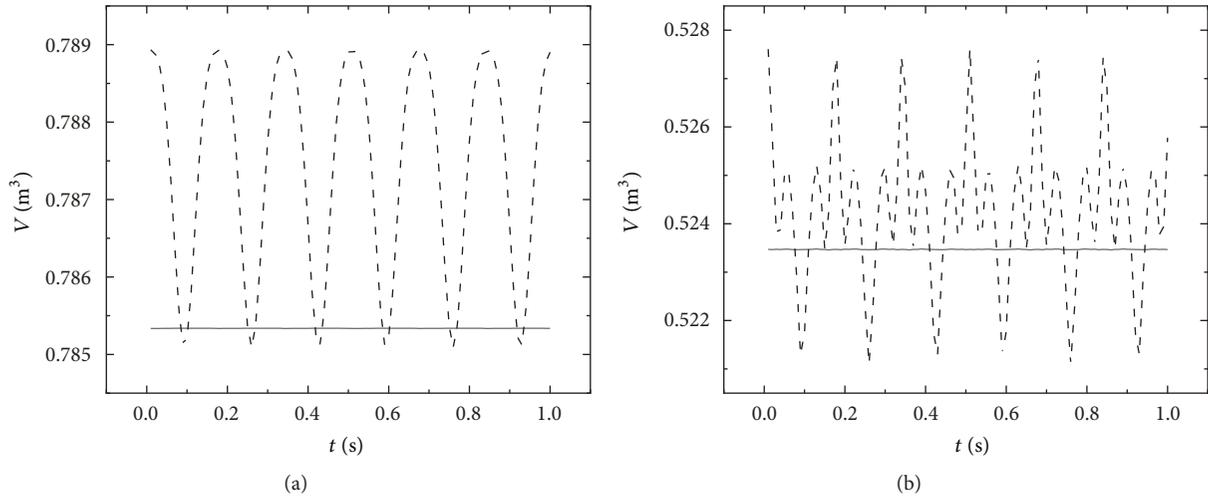


FIGURE 5: Evolution of predicted volume of a moving particle with time: (a) 2D; (b) 3D. solid line: the present numerical results; dash line: numerical results with Gallier's method [14].

method, showing interaction-volume based particle indicator function  $I(\mathbf{x})$  is more accurate.

It can be found from Tables 1 and 2 that the relative error is less than  $5e-5$  if the subdivision level is greater than or equal to 2 for the grid spacing with  $d/20$ . The subdivision level is set to 2 in the later tests, which is a good compromise between computational load and accuracy.

Figure 5 shows the evolution of particle volume with time obtained by two methods when the particle moves on the background mesh in 2D and 3D, respectively. It is

not difficult to find that approximated Heaviside function method is not conservative because the results significantly oscillate with the particle motion, while our method in this paper is conservative and the results are stable.

**4.2. Flow around a Cylinder.** The flow around a cylinder is a classical hydrodynamic problem. A large number of physical and numerical experiments have been carried out previously. The accuracy and validity of this method in the stationary boundary are verified by the flow around a circular

TABLE 3: Comparison of the present computation of drag coefficient ( $C_d$ ) with other results for flow past a cylinder.

$Re$	Present			Gallier et al. [14] (Numerical)	$C_d$		Takami et al. [23] (Numerical)
	Case1	Case2	Case3		Nishioka et al. [21] (Experimental)	Rajani et al. [22] (Numerical)	
10	2.991	3.037	3.049	3.116	2.74	2.839	2.754
20	2.126	2.164	2.174	2.236	1.95	2.059	2.003
40	1.573	1.604	1.611	1.668	1.49	1.539	1.536
100	1.302	1.321	1.328	1.376	—	1.335	—
200	1.246	1.335	1.381	1.402	—	1.337	—

TABLE 4: The non-dimensional length of recirculation zone ( $L_w/d$ ) at  $Re = 20, 40$ .

$Re$	Present			$L_w/d$		Ye et al. [25] (Numerical)
	Case1	Case2	Case3	Takami et al. [23] (Numerical)	Silva et al. [24] (Numerical)	
20	0.95	1.05	1.07	0.932	1.05	0.92
40	2.03	2.25	2.31	2.325	2.54	2.27

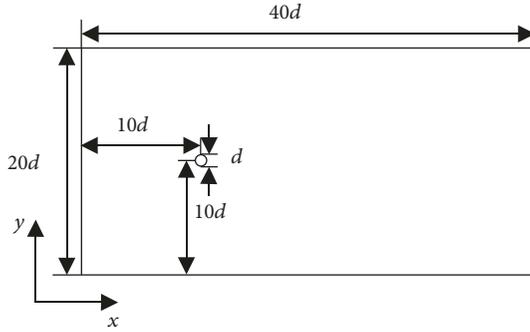


FIGURE 6: Sketch of the computational domain for flow around a circular cylinder.

cylinder with different Reynolds numbers. In this example, the computational domain is a  $40d \times 20d$  rectangular domain (diameter of the cylinder:  $d = 0.025$  m). The centre of the cylinder is located at  $(10d, 10d)$ , as shown in Figure 6. The fluid flows into the left inlet and out of the right outlet, with the inflow velocity,  $\mathbf{u}$  (different for different  $Re$ ), fluid density,  $\rho_f = 1.293 \text{ kg}\cdot\text{m}^{-3}$ , and kinetic viscosity coefficient,  $\nu = 1.5 \times 10^{-5} \text{ m}^2\cdot\text{s}^{-1}$ . The inlet flow velocity is specified, and the pressure is zero-gradient, while at the outlet, Neumann boundary condition is specified for velocity and Dirichlet boundary condition for pressure. The walls at top and bottom are set as the symmetric boundary condition. The Reynolds numbers,  $Re (= \rho_f d |\mathbf{u}| / \nu)$ , of 10, 20, 40, 100, and 200, are selected as test conditions in this example, with the corresponding inlet velocities of  $6 \times 10^{-3} \text{ m}\cdot\text{s}^{-1}$ ,  $1.2 \times 10^{-2} \text{ m}\cdot\text{s}^{-1}$ ,  $2.4 \times 10^{-2} \text{ m}\cdot\text{s}^{-1}$ ,  $6 \times 10^{-2} \text{ m}\cdot\text{s}^{-1}$ , and  $0.12 \text{ m}\cdot\text{s}^{-1}$ , respectively. In order to reduce the computational cost, a refinement mesh is used within a  $5d \times 5d$  square, whose centre is the same as the cylinder's and edges are parallel to the computational domain's, and  $d/5 \times d/5$  meshes are used at the rest of the area in this example. Three levels of grid spacings are employed:

$d/10$  for Case 1,  $d/20$  for Case 2, and  $d/30$  for Case 3, respectively. The corresponding time steps are set so that the Courant number  $Co (= U_0 d / \Delta t)$  maintains around 0.2.

Table 3 gives the drag coefficients  $C_d$ , at different  $Re$  and grid spacings, at the ultimately stable or cyclically stable flow around a circular cylinder. The drag coefficients are obtained from the following equation.

$$C_d = \frac{2 |\mathbf{F}^{hyd}|}{(\rho_f d |\mathbf{u}|^2)} \quad (20)$$

The computed  $C_d$  are convergent with the decrease of grid spacing. For the purpose of comparison, this table also gives the numerical results from the method in [14] (Gallier) and the experimental and numerical results in [21–23]. It can be seen from the table that our results are better almost in all cases than Gallier's method and closer to the experimental and calculated results in the references.

Figure 7 shows the streamlines at  $Re$  of 10, 20, and 40 of Case 3 when the flow becomes eventually stable. It can be seen that stable recirculation zones (whirlpool vortex) are generated behind the cylinders in all three cases, and the length,  $L_w$ , of the recirculation zone increases with the increasing  $Re$ . Table 4 shows the comparison of the length,  $L_w/d$  ( $d$  is the diameter of the cylinder), of recirculation zone calculated by our method at  $Re$  of 20 and 40 for different grid spacings with those in [23–25]. The calculated results are convergent with the refinement of the mesh and in good agreement with those in the references.

Figure 8 shows the velocity contour of cyclically stable flow at  $Re$  of 100 and 200 of Case 3. In both cases, the whirlpool vortex behind the cylinder begins to regularly shed and forms a regular vortex street. Table 5 shows the comparison of Strouhal number ( $S_t = fd / |\mathbf{u}|$ , where  $f$  is the vortex shedding frequency) calculated by this method in both cases with those in [7, 26, 27]. The calculated results are converted with the decrease of grid spacing and are in good agreement with those in the references.

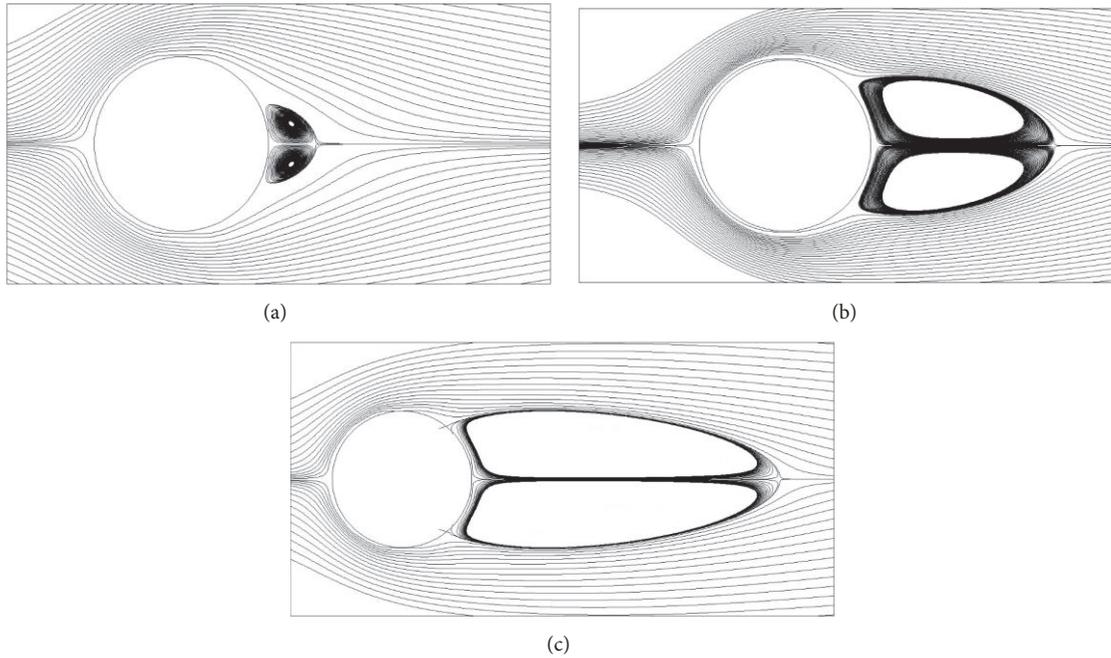


FIGURE 7: Streamlines for flow around a circular cylinder of Case 3 at  $Re =$  (a) 10; (b) 20; (c) 40.

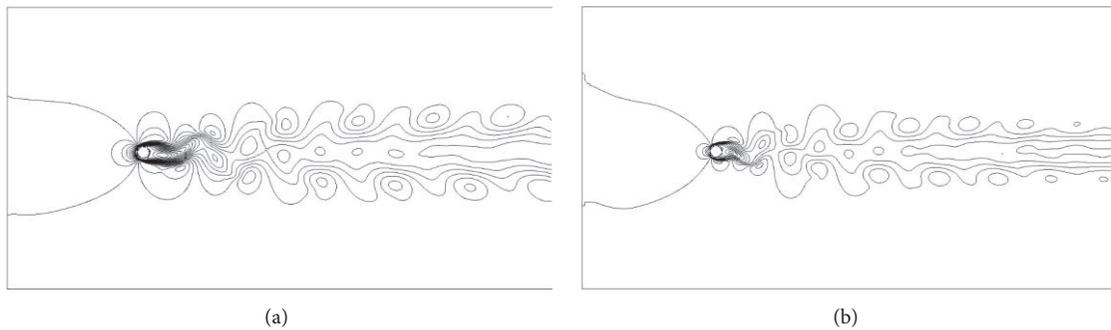


FIGURE 8: Velocity contour for flow around a circular cylinder of Case 3 at  $Re =$  (a) 100, (b) 200.

TABLE 5: Comparison of the numerical predicted Strouhal number ( $St$ ) with different methods for flow around a circular cylinder.

$Re$	$St$					
	Case1	Present Case2	Case3	Liu et al. [26]	Uhlmann [7]	Tseng and Ferziger [27]
100	0.162	0.165	0.166	0.164	0.169	0.164
200	0.189	0.193	0.194	0.192	—	—

The simulation results in this example indicate that, for the static boundary problem, the method in this paper is of good accuracy in a wide range of  $Re$ .

**4.3. Single Spherical Particle in a Creeping Shear Flow.** In this section, a single force-free torque-free sphere in a linear creeping shear flow is considered to verify the feasibility of this method in dealing with sphere rotation. A spherical particle of diameter  $d = 0.1$  m is suspended at the centre of a cubic Couette cell, as shown in Figure 9. The particle is located at  $x = 0$  and the cell size is  $L = 10d$ . A velocity

of  $\gamma L/2$  and  $-\gamma L/2$  in the flow direction is assigned to the top and bottom boundaries (the dark ones), respectively, where  $\gamma = 6.71 \times 10^{-3} s^{-1}$  is the shear rate. And for the rest of the boundaries, periodic conditions are used. The kinematic viscosity of the flow is  $\nu = 10^{-3} m^2 s^{-1}$ . Three levels of grid spacings are used:  $\Delta = d/5$  for Case 1,  $\Delta = d/7.5$  for Case 2, and  $\Delta = d/10$  for Case 3. And the corresponding time steps are 0.01s, 0.005s, and 0.004s, respectively.

In a creeping shear flow, the effect of convection term is so little that it can be neglected, and unsteady Stokes equations are solved with respect to time-stepping. Eventually, the

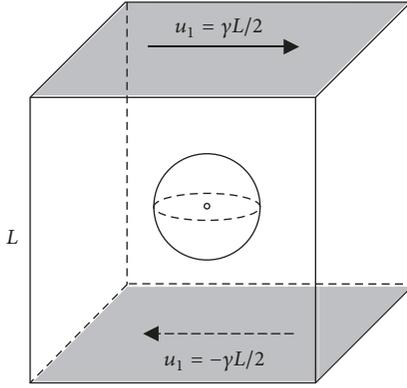


FIGURE 9: Sketch of the computational domain for a single spherical particle in creeping shear flow.

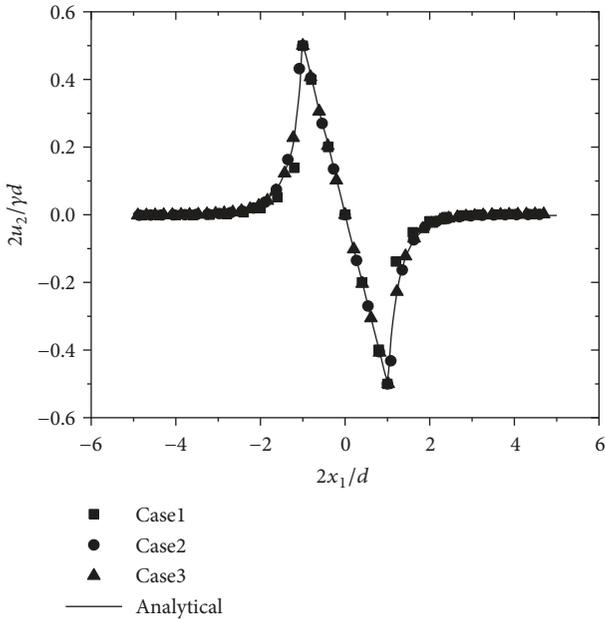


FIGURE 10: Evolution of the centreline fluid velocity  $2u_2/\gamma d$  with position  $2x_1/d$  for a single sphere located at the origin ( $x = 0$ ).

whole system reaches a steady state, and the particle is rotating in a constant angular velocity, which is available of analytical solutions [14].

Figure 10 shows a comparison between the profile of the computed nondimensional velocity  $2u_2/\gamma d$  on different grid spacings on the centreline ( $x_2 = 0$ ) of the shear plane ( $x_3 = 0$ ), where the subscript represents the direction of the flow, and the analytical solution [28]. For the coarsest mesh, Case 1, the computed results near the outside region of the particle surface are slightly lower than the analytical solution. With the refinement of the mesh, the results become more and more close to the analytical solution both inside and outside the particle. The linear profile inside the particle ( $2|x_1|/d < 1$ ) is due to the rigid body motion constraint of the particle rotation and a velocity  $2|u_2|/\gamma d = 0.5$  is found at the surface of the particle ( $2|x_1|/d = 1$ ). The rotational velocities  $\Omega_3/\gamma$  of

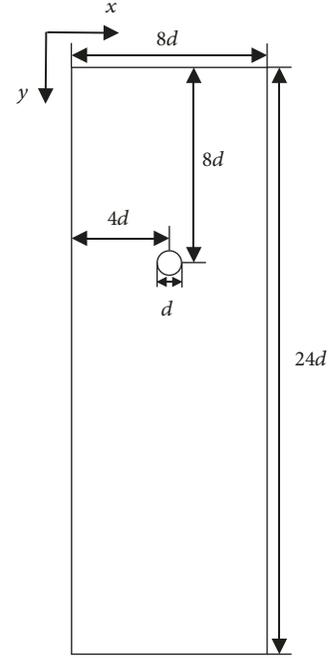


FIGURE 11: Sketch of the computational domain for a circular particle settling in a closed container.

all three cases are  $-0.4990$ ,  $-0.4995$ , and  $-0.4996$ , respectively, and gradually approach the theoretical value  $-0.5$ . The particle translational velocity is also solved and turns out to be of the order of  $10^{-8}$ , which is acceptable with respect to the expected value 0.

The simulation results in this case show that this method is feasible for the simulation of spherical particle rotation.

#### 4.4. Sedimentation of a Circular Particle in a Closed Container.

The sedimentation process of a circular particle in a closed container is simulated in this case to verify the feasibility of this method in dealing with moving boundary problems. The computational model is shown in Figure 11. The computational domain is  $8d \times 24d$  (diameter of a particle:  $d = 2.5 \times 10^{-3}$  m), and the initial position of the particle is  $(4d, 8d)$ . The fluid is stationary at the initial time. The circular particle settles by gravity, whose density is  $\rho_p = 1250 \text{ kg}\cdot\text{m}^{-3}$ . The dynamic viscosity and density of the fluid are  $\mu = 0.01 \text{ kg}\cdot\text{m}^{-1}$  and  $\rho_f = 1000 \text{ kg}\cdot\text{m}^{-3}$ , respectively. For a closed container, nonslip solid walls conditions are prescribed at the surrounded boundaries.

In this example, the sedimentation process of the particle is simulated. At the beginning, the particle begins to settle by gravity, which is more than buoyant force. With the increase of the settling velocity, the drag force on the particle increases gradually and the settling acceleration decreases gradually. Finally, the settling velocity reaches a constant value due to the equilibrium of three forces, gravity, buoyancy, and drag force. For the purpose of convergence test, a series of grid spacings are considered, as shown in Table 6, with a series of properly chosen time steps to maintain the Courant number around 0.2.

TABLE 6: A series of grid spacings with time steps for different cases.

Case	1	2	3	4	5	6	7
Grid spacing/m	$d/12.5$	$d/18.75$	$d/25$	$d/31.25$	$d/37.5$	$d/43.75$	$d/50$
Time step/s	$7.5e-4$	$5e-4$	$3.75e-4$	$3e-4$	$2.5e-4$	$2.14e-4$	$1.975e-4$

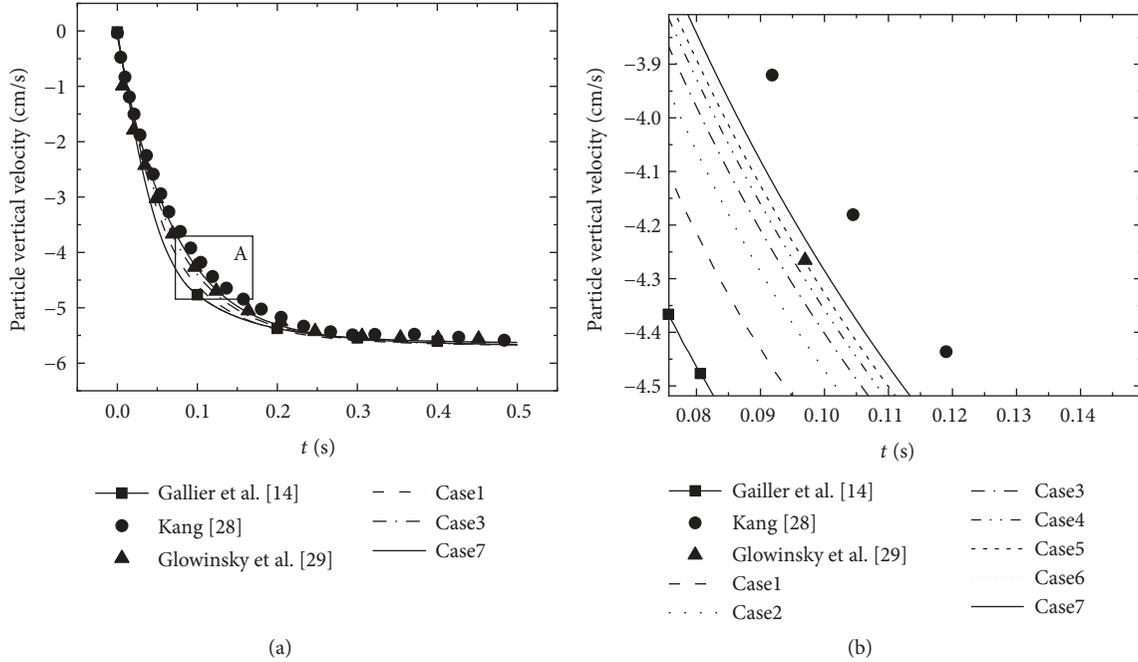


FIGURE 12: Evolution of particle vertical velocity with time (a). (b) is the enlarged view of area A in (a).

Figure 12(a) gives the evolution of particle vertical velocity with time, and Figure 12(b) is the enlarged view of area A in Figure 12(a). It is clear that the computed results are convergent with the decrease of grid spacing. For the purpose of comparison, Kang's method [29] and Glowinski's method [30] are also covered. It can be found that the method presented in this paper has closer results in all cases to Kang's and especially Glowinski's methods than Gallier's (configuration is the same as case 2). Figure 13(a) shows the evolution of particle vertical position with time, and Figure 13(b) is the enlarged view of area A in Figure 13(a). Similar to Figure 12, the results are convergent with the decrease of grid spacing, and all of them are closer to Kang's method than Gallier's method. Glowinski did not present the evolution of particle vertical position with time in his paper, but the vertical velocity obtained by this method is consistent with that in his paper, so the vertical position predicted in this paper should not be very different from that calculated by Glowinski's method.

The simulation results illustrate that the method presented in this paper can deal with the moving boundary problems very well.

**4.5. In-Line Oscillation of a Circular Cylinder.** To further verify the feasibility of this method in dealing with moving boundary problems, an in-line oscillation of a circular

cylinder is selected as an example for study. As shown in Figure 14, the computational domain is  $30d \times 20d$  ( $d = 1$  cm is the diameter of the cylinder), the circumference is set to be the second boundary condition with zero velocity gradient, and the initial position of the cylinder is  $(0, 0)$ . The fluid is stationary at the initial time, with the density,  $\rho_f = 1000$   $\text{kg} \cdot \text{m}^{-3}$ . The cylinder is periodically reciprocating according to the following expression

$$X_c(t) = -A \sin\left(\frac{2\pi t}{T}\right) \quad (21)$$

where  $X_c(t)$  is the abscissa of the centre of the cylinder;  $A$  and  $T$  are amplitude and period, respectively. Reynolds number,  $Re$ , and Keulegan-Carpenter number,  $KC$ , which determine the characteristics of the flow, are calculated as follows

$$Re = \frac{U_{c,max}d}{\nu} \quad (22)$$

$$KC = \frac{U_{c,max}T}{d} \quad (23)$$

where  $U_{c,max}$  is the maximum velocity of the cylinder during oscillation and  $\nu$  is the coefficient of kinematic viscosity of the fluid.  $Re = 100$ ,  $KC = 5$  are selected in this example, similar to the experimental conditions selected by Dutsch et al. [18]. The grid spacing is  $d/20$  in size.

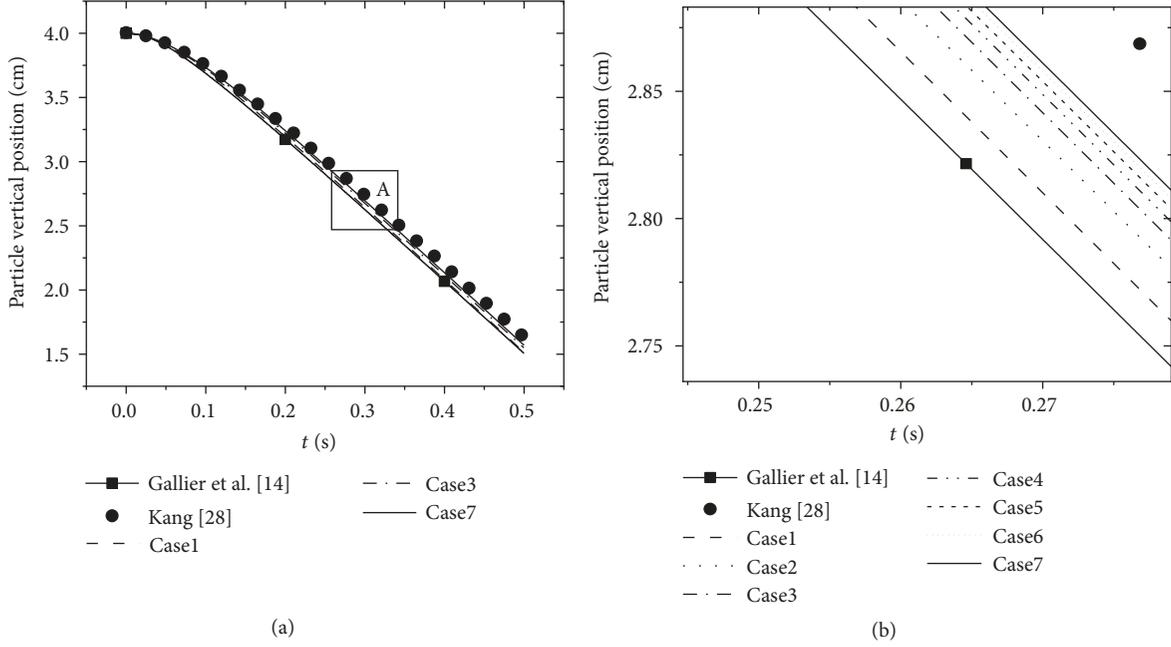


FIGURE 13: Evolution of particle vertical position with time (a). (b) is the enlarged view of area A in (a).

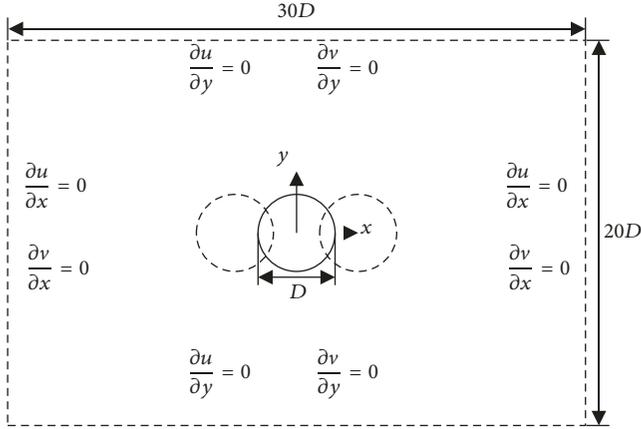


FIGURE 14: Sketch of the computational domain for a circular cylinder oscillating in-line in a quiescent fluid.

Figure 15 shows the stream force coefficient obtained by different methods within a period after stabilization.  $C_x$  is calculated according to the following formula

$$C_x = \frac{2F_x}{(\rho_f U_{c,max}^2 d)} \quad (24)$$

where  $F_x$  is the horizontal resultant force on the cylinder.  $F_x$  is calculated according to the following formula

$$F_x = -\rho_f \int_P \lambda_x dx = -\rho_f \int_{D_p} I(x) \lambda_x dx \quad (25)$$

For the purpose of comparison, the results from Gallier's method and experimental data from Dutsch are also given

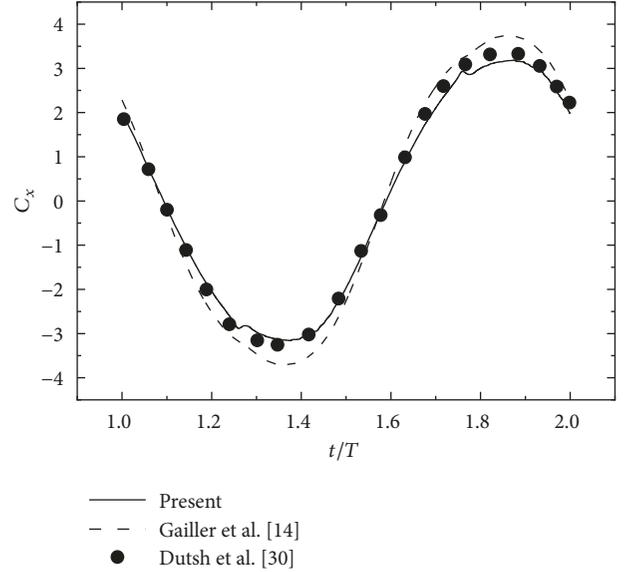


FIGURE 15: Comparison of the streamwise force coefficient.

in the figure. It can be seen from the figure that although Gallier's method provides smoother calculation results, the absolute values at the crest and trough are larger than the experimental values; on the contrary, in spite of small oscillation of the results from this method at  $1.25T$  and  $1.75T$ , the results are closer to the experimental values mainly due to the fact that the particle indicator function  $I(\mathbf{x})$  of Gallier's method is absolutely smooth at the interface of two phases but cannot guarantee the conservation of the integral term in Formula (25), resulting in the smooth change of  $C_x$  with time,

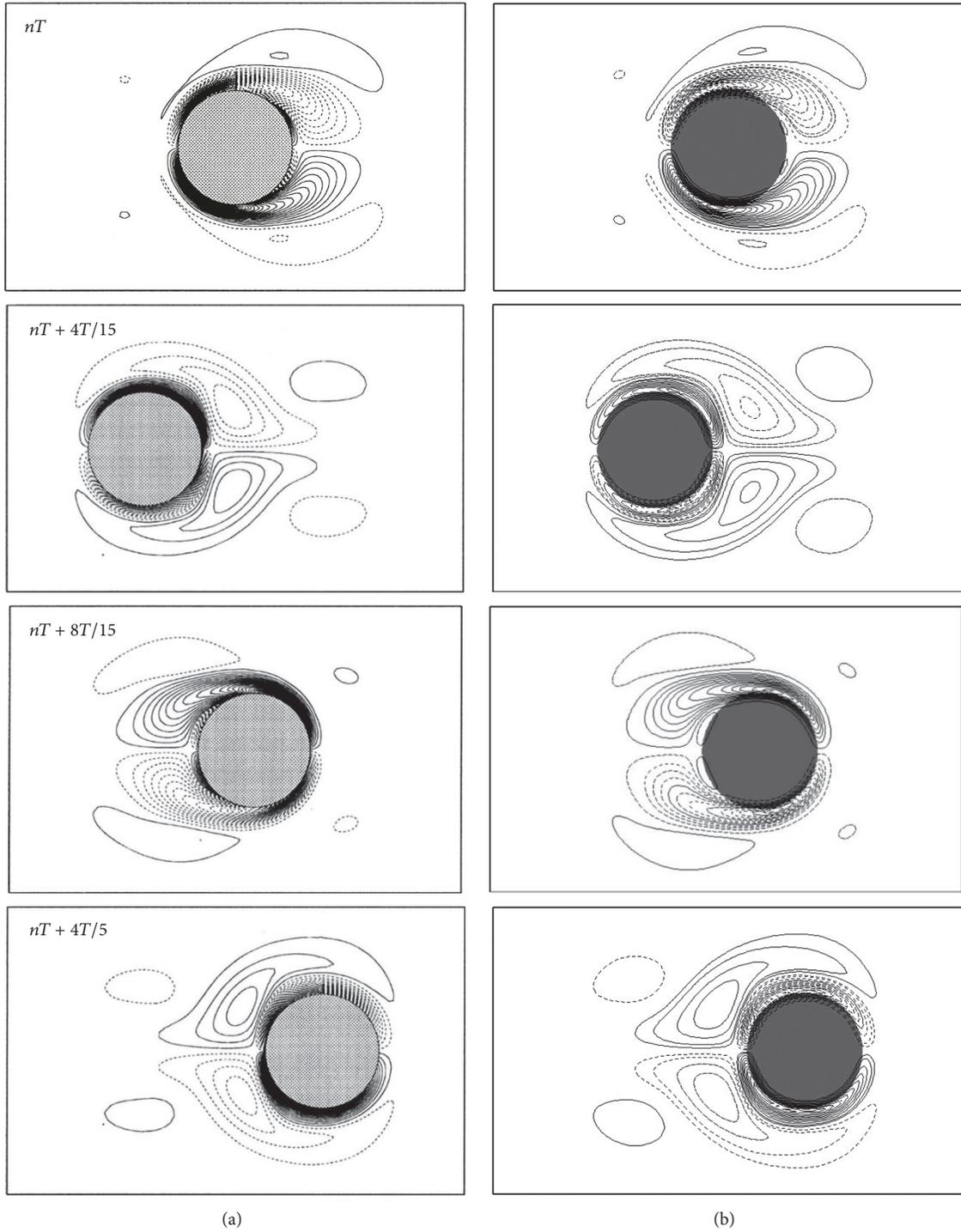


FIGURE 16: Vorticity contour (negative values dashed) at different phase time: (a) numerical results of Dutsch [18]; (b) the present numerical results.

but inaccurate. However,  $I(x)$  from this method is absolutely conservative, but there are some discontinuous changes at the interface with the motion of the cylinder, resulting in small oscillations at  $1.25T$  and  $1.75T$  when the cylinder switches its direction. Yet the results obtained by this method agree with Dutsch's experimental results.

Figure 16 shows the vorticity around the cylinder at different phase times within a period after stabilization. It can be spotted that the results in this paper are in good agreement with those from Dutsch et al.

Figures 17, 18, and 19 show the profiles of the component  $u$  in the  $x$  direction and  $v$  in the  $y$  direction at  $x = -6, 0, 6$ , and

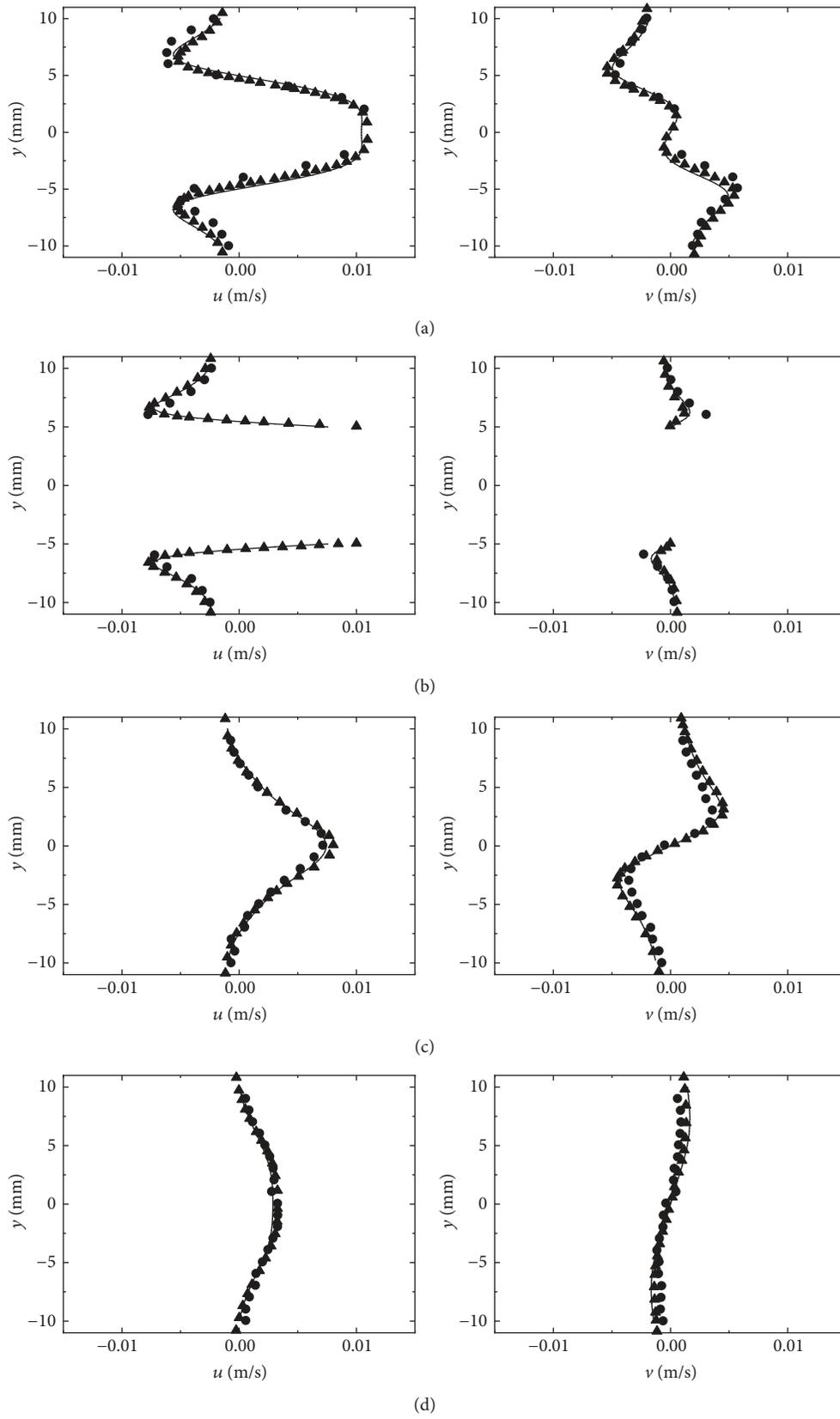


FIGURE 17: Comparison of the velocity components at four cross sections  $x =$ (a) -6, (b) 0, (c) 6, (d) 12mm at  $t = nT + T/2$ . Solid line: the present numerical results;  $\bullet$ : experimental data [18];  $\blacktriangle$ : numerical results [18].

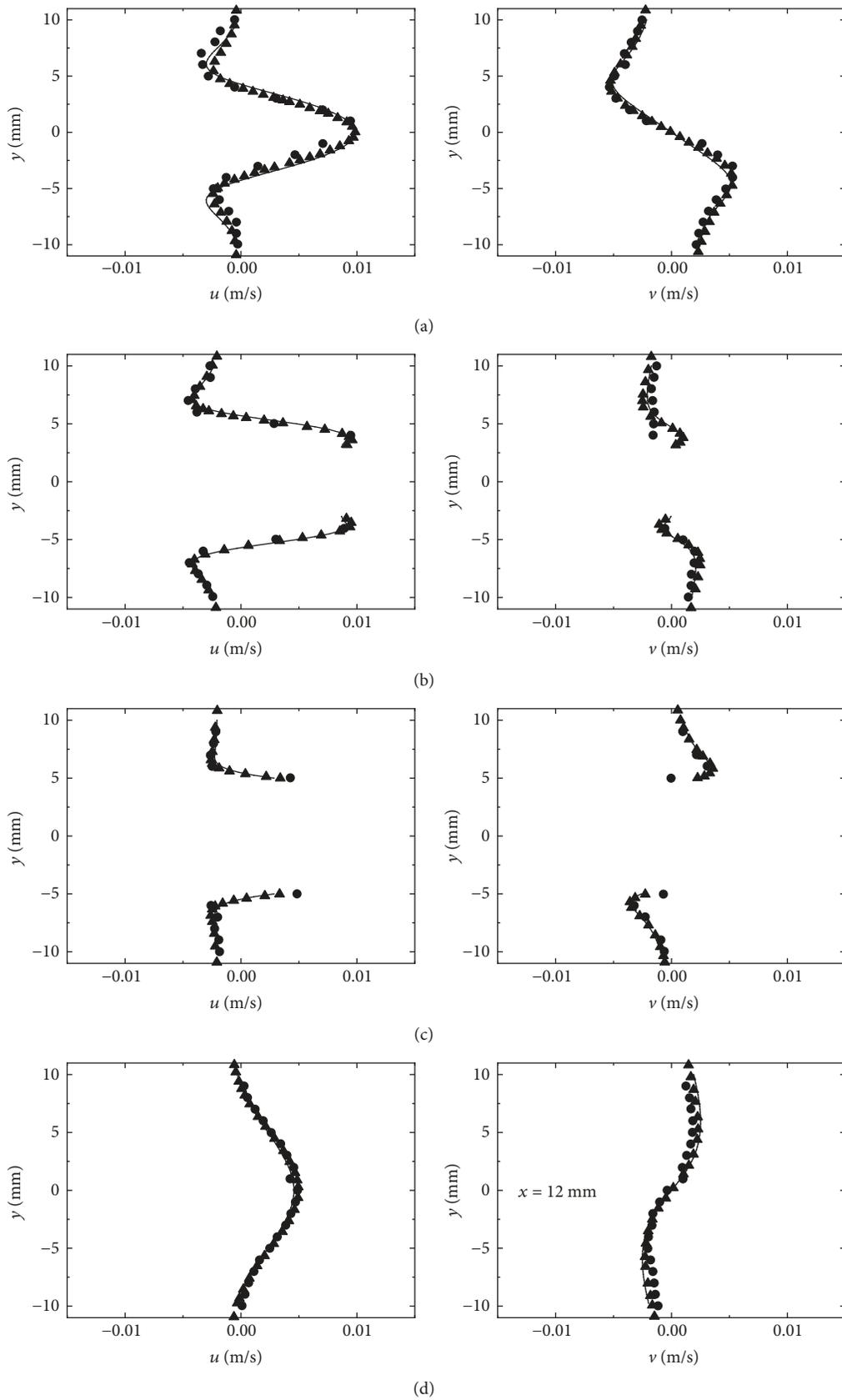


FIGURE 18: Comparison of the velocity components at four cross sections  $x=(a)$  -6 mm,  $(b)$  0 mm,  $(c)$  6 mm,  $(d)$  12 mm at  $t = nT + 7T/12$ . Solid line: the present numerical results; ●: experimental data [18]; ▲: numerical results [18].

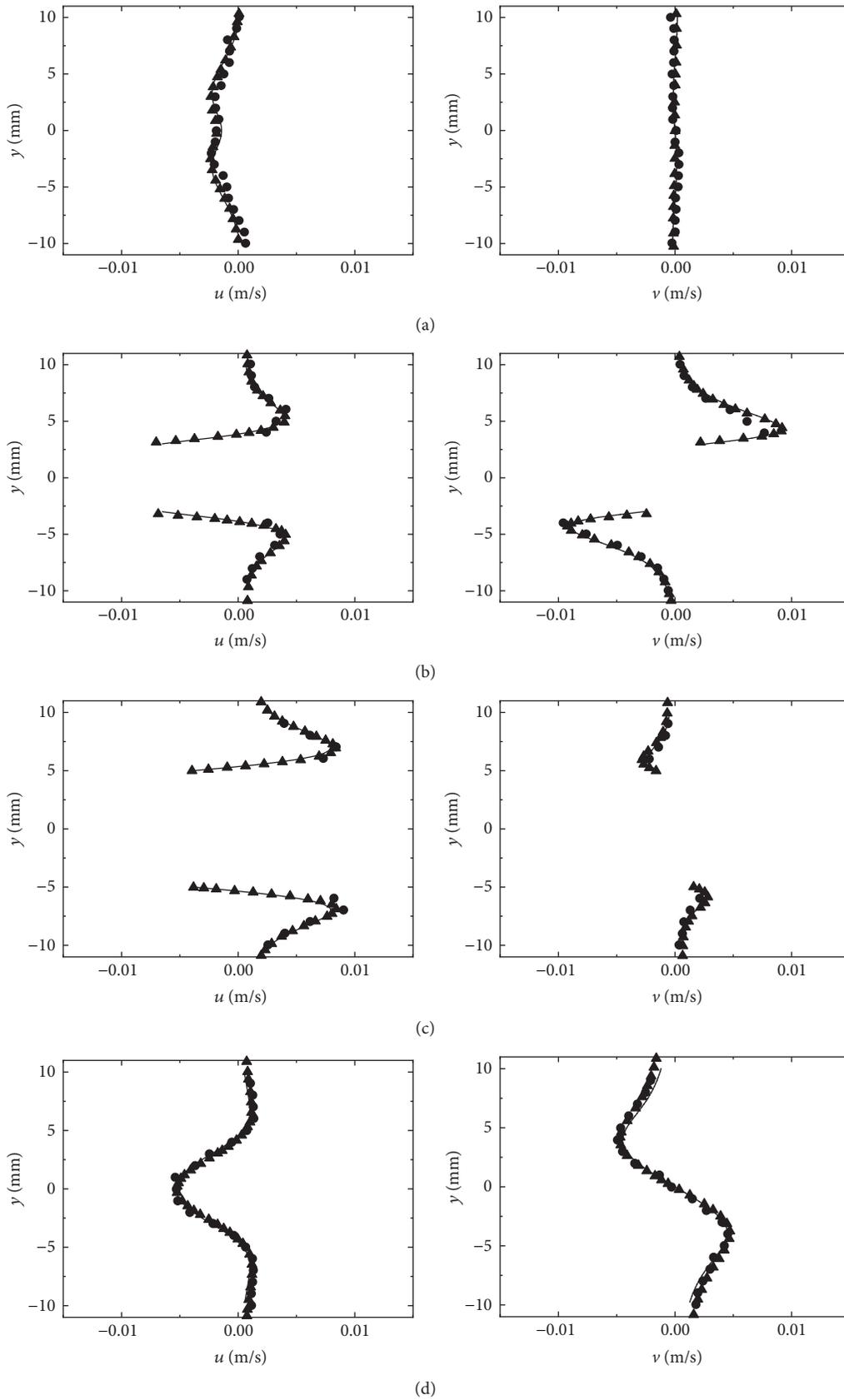


FIGURE 19: Comparison of the velocity components at four cross sections  $x=(a)$  -6 mm,  $(b)$  0 mm,  $(c)$  6 mm,  $(d)$  12 mm at  $t = nT + 11T/12$ . Solid line: the present numerical results;  $\bullet$ : experimental data [18];  $\blacktriangle$ : numerical results [18].

12 mm at the phase time  $t = nT + T/2$ ,  $t = nT + 7T/12$ , and  $t = nT + 11T/12$ , respectively. For the purpose of comparison, the experimental data and numerical results from Dutsch are also presented. It can be seen that the results in this paper are in good agreement with the data from Dutsch.

The simulation results from this example further indicate that the method in this paper is feasible in dealing with the moving boundary problems.

## 5. Conclusions

A fictitious domain method based on the particle indicator function from intersection volume is developed in this paper for dealing with moving boundary problems. In this method, the background mesh cells across the interface between fluid and particles are located and subdivided into tetrahedrons. The volume fraction of the particle domain within each cell is accurately computed to achieve high precision of integration calculation in the particle domain, improving the accuracy of the whole method. The integral value of the particle domain calculated by this function is more accurate, which greatly improves the overall calculation accuracy of the fictitious domain method. Volume evaluation based on cells subdivision techniques shows that the technology can accurately calculate the volume of static particles and guarantee the conservation of the calculated integral on the moving particle. The test results of the flow around a cylinder, a single sphere in a creeping shear flow, the sedimentation of a circular particle in a closed container, and in-line oscillation of a circular cylinder show that this fictitious domain method has good accuracy in dealing with stationary boundary and moving boundary problems. This method has better performance in simulation of stationary and moving particle-fluid motion than the method using approximate Heaviside function. This method is more accurate and conservative. It can be adopted to deal with moving and stationary boundary problems.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The project supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China (Grant No. 2016ZX05011001), the National Natural Science Foundation of China (Grant No. 21306145), and Nature Science Basic Research Plane in Shaanxi Province of China (Grant No. 2019JQ-335).

## References

- [1] L. Xu, K. Luo, Y. Zhao, J. Fan, and K. Cen, "Multiscale investigation of tube erosion in fluidized bed based on CFD-DEM simulation," *Chemical Engineering Science*, vol. 183, pp. 60–74, 2018.
- [2] J. Su, Z. Gu, and X. Y. Xu, "Discrete element simulation of particle flow in arbitrarily complex geometries," *Chemical Engineering Science*, vol. 66, no. 23, pp. 6069–6088, 2011.
- [3] W. Ge, L. Wang, J. Xu et al., "Discrete simulation of granular and particle-fluid flows: from fundamental study to engineering application," *Reviews in Chemical Engineering*, vol. 33, no. 6, pp. 551–623, 2017.
- [4] J. L. Escalona, "An arbitrary Lagrangian-Eulerian discretization method for modeling and simulation of reeving systems in multibody dynamics," *Mechanism and Machine Theory*, vol. 112, pp. 1–21, 2017.
- [5] R. Jafari and T. Okutucu-Özyurt, "3D numerical modeling of boiling in a microchannel by arbitrary Lagrangian-Eulerian (ALE) method," *Applied Mathematics and Computation*, vol. 272, pp. 593–603, 2016.
- [6] K. Yang, F. Hong, and P. Cheng, "A fully coupled numerical simulation of sessile droplet evaporation using Arbitrary Lagrangian-Eulerian formulation," *International Journal of Heat and Mass Transfer*, vol. 70, pp. 409–420, 2014.
- [7] M. Uhlmann, "An immersed boundary method with direct forcing for the simulation of particulate flows," *Journal of Computational Physics*, vol. 209, no. 2, pp. 448–476, 2005.
- [8] K. Taira and T. Colonius, "The immersed boundary method: a projection approach," *Journal of Computational Physics*, vol. 225, no. 2, pp. 2118–2137, 2007.
- [9] R. Glowinski, T.-W. Pan, T. I. Hesla, and D. D. Joseph, "A distributed Lagrange multiplier/fictitious domain method for particulate flows," *International Journal of Multiphase Flow*, vol. 25, no. 5, pp. 755–794, 1999.
- [10] J. Su, G. Chai, L. Wang et al., "Pore-scale direct numerical simulation of particle transport in porous media," *Chemical Engineering Science*, vol. 199, pp. 613–627, 2019.
- [11] N. Sharma and N. A. Patankar, "A fast computation technique for the direct numerical simulation of rigid particulate flows," *Journal of Computational Physics*, vol. 205, no. 2, pp. 439–457, 2005.
- [12] J. Su, G. Chai, L. Wang et al., "Direct numerical simulation of pore scale particle-water-oil transport in porous media," *Journal of Petroleum Science and Engineering*, vol. 180, pp. 159–175, 2019.
- [13] Z. Yu and X. Shao, "Direct numerical simulation of particulate flows with a fictitious domain method," *International Journal of Multiphase Flow*, vol. 36, no. 2, pp. 127–134, 2010.
- [14] S. Gallier, E. Lemaire, L. Lobry, and F. Peters, "A fictitious domain approach for the simulation of dense suspensions," *Journal of Computational Physics*, vol. 256, pp. 367–387, 2014.
- [15] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, and J. Périaux, "A distributed lagrange multiplier/fictitious domain method for flows around moving rigid bodies: Application to particulate flow," *International Journal for Numerical Methods in Fluids*, vol. 30, no. 8, pp. 1043–1066, 1999.
- [16] S. Haeri and J. S. Shrimpton, "On the application of immersed boundary, fictitious domain and body-conformal mesh methods to many particle multiphase flows," *International Journal of Multiphase Flow*, vol. 40, pp. 38–55, 2012.
- [17] R. Rabbittz, "Fast collision detection of moving convex polyhedra," in *Graphics Gems IV*, pp. 83–109, Academic Press, London, UK, 1994.
- [18] H. Dütsch, F. Durst, S. Becker, and H. Lienhart, "Low-reynolds-number flow around an oscillating circular cylinder at low

- keulegan-carpenter numbers,” *Journal of Fluid Mechanics*, vol. 360, pp. 249–271, 1998.
- [19] R. I. Issa, “Solution of the implicitly discretised fluid flow equations by operator-splitting,” *Journal of Computational Physics*, vol. 62, no. 1, pp. 40–65, 1986.
- [20] J. Su, Z. Gu, C. Chen, and X. Xu, “A two-layer mesh method for discrete element simulation of gas-particle systems with arbitrarily polyhedral mesh,” *International Journal for Numerical Methods in Engineering*, vol. 103, no. 10, pp. 759–780, 2015.
- [21] M. Nishioka and H. Sato, “Measurements of velocity distributions in the wake of a circular cylinder at low Reynolds numbers,” *Journal of Fluid Mechanics*, vol. 65, no. 1, pp. 97–112, 1974.
- [22] B. N. Rajani, A. Kandasamy, and S. Majumdar, “Numerical simulation of laminar flow past a circular cylinder,” *Applied Mathematical Modelling*, vol. 33, no. 3, pp. 1228–1247, 2009.
- [23] H. Takami and H. B. Keller, “Steady two-dimensional viscous flow of an incompressible fluid past a circular cylinder,” *Physics of Fluids*, vol. 12, no. 12, pp. II-51–II-56, 1969.
- [24] A. L. E. Silva, A. Silveira-Neto, and J. Damasceno, “Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method,” *Journal of Computational Physics*, vol. 189, no. 2, pp. 351–370, 2003.
- [25] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, “An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries,” *Journal of Computational Physics*, vol. 156, no. 2, pp. 209–240, 1999.
- [26] C. Liu, X. Zheng, and C. H. Sung, “Preconditioned multigrid methods for unsteady incompressible flows,” *Journal of Computational Physics*, vol. 139, no. 1, pp. 35–57, 1998.
- [27] Y.-H. Tseng and J. H. Ferziger, “A ghost-cell immersed boundary method for flow in complex geometry,” *Journal of Computational Physics*, vol. 192, no. 2, pp. 593–623, 2003.
- [28] G. K. Batchelor and J. T. Green, “The hydrodynamic interaction of two small freely-moving spheres in a linear flow field,” *Journal of Fluid Mechanics*, vol. 56, no. 2, pp. 375–400, 1972.
- [29] S. K. Kang, *Immersed Boundary Methods in The Lattice Boltzmann Equation for Flow Simulation*, Texas A&M University, College Station, TX, USA, 2012.
- [30] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, and J. Periaux, “A distributed Lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: application to particulate flow,” *Computer Methods Applied Mechanics and Engineering*, vol. 184, no. 2-4, pp. 241–267, 2000.

