Paper:
Wang, M., Feng, Y., Owen, D. & Qu, T. (2018). A novel algorithm of immersed moving boundary scheme for fluid-particle interactions in DEMLBM. *Computer Methods in Applied Mechanics and Engineering*
http://dx.doi.org/10.1016/j.cma.2018.12.001

Accepted Manuscript

A novel algorithm of immersed moving boundary scheme for fluid-particle interactions in DEMLBM

Min Wang, Y.T. Feng, D.R. J. Owen, T.M. Qu

PII: S0045-7825(18)30598-X
DOI: https://doi.org/10.1016/j.cma.2018.12.001
Reference: CMA 12192

To appear in: *Comput. Methods Appl. Mech. Engrg.*

Received date : 25 August 2018
Revised date : 30 November 2018
Accepted date : 1 December 2018

Please cite this article as: M. Wang, Y.T. Feng, D.R.J. Owen et al., A novel algorithm of immersed moving boundary scheme for fluid-particle interactions in DEMLBM, *Computer Methods in Applied Mechanics and Engineering* (2018), https://doi.org/10.1016/j.cma.2018.12.001

Highlights (for review)

1) The novel boundary tracing procedure proposed in this paper is robust and efficient. Compared to our previous linear searching algorithm, the computing cost is reduced by half.
2) Although only circular particles are used to examine the novel searching method, the algorithm itself is universal and can be easily implemented for polygons.
3) The Gaussian quadrature for computing solid nodal ratio is of high accuracy and efficiency. Compare to another fast method, Monte Carlo, to achieve the same and high accuracy only a few points are needed, while the latter needs at least a thousand.
4) The IMB scheme is more stable than the IBM scheme in terms of the calculated hydrodynamic force, and therefore is more robust to simulate problems with a large number of particles and/or more complex fluid flow patterns.

A Novel Algorithm of Immersed Moving Boundary Scheme for Fluid-Particle Interactions in DEMLBM

Min Wang[1], Y.T. Feng[2]*, D.R.J. Owen[2], T.M. Qu[2]

1 T-3 Fluid Dynamics and Solid Mechanics Group, Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

2 Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Swansea, Wales SA1 8EP, UK

* Corresponding e-mail: y.feng@swansea.ac.uk

Abstract

This paper presents an efficient and accurate Immersed Moving Boundary (IMB) algorithm for solving fluid-particle interactions in the framework of the Lattice Boltzmann Method (LBM). Although the IMB scheme has been widely employed in many fluid-particle coupling problems in a wide range of applications, the algorithm of its implementation, especially in identifying both fluid and solid boundary nodes for particles, is seldom reported. Besides, the computational cost of handling fluid-particle coupling is very expensive. To provide a bridge between theory and application and improve the computing efficiency of IMB, a novel boundary tracing procedure and an efficient method for computing the solid nodal ratio using Gaussian quadrature are proposed in this paper. Both accuracy and efficiency of the proposed algorithm are examined by two benchmark tests. It is also found that the IMB scheme are more efficient and stable compared to another widely used the Immersed Boundary Method (IBM) in LBM.

Keywords

Fluid-solid interaction; Lattice Boltzmann Method; Discrete Element Method; Immersed Moving Boundary; Immersed Boundary Method; Boundary Tracing

1 Introduction

The fluid-particle interaction is a very common issue in chemical engineering, fluid mechanics, geomechanics, computational biomechanics and many other fields. Problems involving the fluid-particle interaction include, e.g., gas/liquid solid fluidised bed, particle transportation in fluid, soil erosion and the flow of blood in the heart, just to name a few. Because of its complexity and significance, fluid-particle systems involving the complex fluid-particle interaction have been extensively investigated since the 1980s.

Most of the proposed methods for fluid-particle systems can be divided into two categories according to the framework of fluid dynamics. One is the conventional Computational Fluid Dynamics (CFD) based techniques in which the fluid is described by the Navier-Stokes equations. Among these methods, the Finite Volume Method (FVM) [1-3] and Finite Element Method (FEM) [4-7] can effectively simulate the behaviour of a small number of particles immersed in fluid. However, for a fluid-particle system involving a large number of particles, these two techniques have to continuously generate geometrically adapted meshes due to the motion of particles. This is obviously very computationally expensive, particularly for the three-dimensional modelling. In order to overcome this difficulty, a local-average method which considers the effect of the presence of particles on the fluid in terms of local porosity was adopted and successfully employed in the coupled CFD and particle methods such as the Discrete Element Method (DEM) [8-13]. At each time step, the hydrodynamic forces acting on individual particles in a fluid cell are calculated first by an empirical local-average method, and the values are then summed to the computational cell of the fluid.

Another Navier-Stokes-based technique for the fluid phase is the Finite Difference Method (FDM) [14-16]. In this method the fluid-particle interaction is resolved by the Immersed Boundary (IB) technique proposed by Peskin [17, 18]. The fluid phase is represented by fixed non-adaptive meshes and the boundaries of moving particles are represented by a set of Lagrangian nodes. The basic idea of the IB scheme is to treat the particle boundary as deformable body with high stiffness. A small distortion of the particle boundary caused by the fluid-particle interactions will generate a force that tends to restore the particle to its original shape. Boundary deformation is calculated by comparing the boundary point and the reference point that undergoes rigid body motions with particles. The challenge of this method is the proper determination of the stiffness of springs used for calculating hydrodynamic forces. Later, a direct-forcing IB approach was introduced in [63]. In this approach, the body force term is directly deduced from the momentum equation by setting the velocity at IB points to the desired velocity using interpolation/distribution functions. An improved direct-forcing IB approach is reported in [64].

The second category is the Lattice Boltzmann Method (LBM) based techniques where the fluid phase is treated as an assembly of fluid particle packages whose movement is governed by the Lattice Boltzmann Equation. Each particle package carries mass and momentum. In the 1990s, the LBM was successfully applied to solve fluid-particle interaction problems [19-21]. In this method, the modified bounce-back rule [19, 22-24] was used to achieve the no-slip condition at the fluid-particle interface. Each particle is divided into a large number of solid nodes by fluid grids. The fluid boundary nodes, exterior to the particle surface, and the solid boundary nodes, interior to the particle surface, are assumed to be connected by links. The particle boundary is represented by the middle points of the links. Clearly, the stepwise lattice representation of the surface of a circular particle is neither accurate nor smooth unless a sufficiently small grid side is employed. More seriously, when a particle is in motion, its boundary nodes will continually change, but in an 'on-off' fashion, which has a serious impact on the smoothness of the computed hydrodynamic forces [25-27]. In order to resolve the aforementioned problem, three approaches have been proposed. The first one is the so-called interpolation-based approach [28-32]. It is reported, however, that the interpolation routines used to solve the distribution functions near the curved boundary result in a loss of mass conservation, which reduces the accuracy of the computed momentum transfer at the boundary [31, 33]. Later, the interpolation-based approach was improved by treating curved boundaries using an appropriate local refinement grid technique [33].

The second approach is the Immersed Moving Boundary (IMB) scheme proposed by Noble and Torczynski [34]. The hydrodynamic forces at the moving boundary are accomplished by introducing an additional collision term to nodes covered partially or fully by the solid particle and a weighting function involving the solid fraction within a computational cell. Because of its more accurate representation of particle boundary, enhanced computational stability and reasonable efficiency, the IMB has been widely used in the coupled DEM-LBM technique where thousands of particles immersed in fluid can be considered without difficulties [26, 35-40]. Recently, an IMB method with modified weighting function was proposed to improve the drag and permeability invariant [62].

The last approach is the IB [17, 18] based technique. The IB scheme was introduced to the Lattice Boltzmann Method by Feng and Michaelides [25] and examined by others [41-43]. It is indicated that in the initial IB-LBM the non-slip boundary condition is not fully enforced and the choice of the spring constant is arbitrary. Recently, Shu et al. [42] proposed an implicit velocity correction based IB-LBM in which the velocity correction should be determined in such a way that the velocity at the boundary interpolated from the corrected velocity field satisfies the non-slip boundary condition. It was then improved in terms of accuracy of hydrodynamic forces [44, 45] and computational efficiency [46, 47]. The computational accuracy of hydrodynamic forces applied to solid particles depends on determination of the number of boundary nodes for the solid particle. If the number exceeds a critical value, this method may suffer from a non-convergence problem. If the number is less, the simulation accuracy of the fluid cannot be guaranteed.

Due to the rapid development of DEM-LBM and its application in fluid-particle coupling, the IMB scheme has been attracting more and more attention. However, in previous studies, only a few literatures have compared the accuracy of IMB in terms of moving particle-fluid interactions with other existing methods [53, 58], e.g. the IBM scheme. Particularly, the detailed algorithm on effectively identifying fluid and solid lattice nodes in IMB was seldom reported [59]. Therefore, this paper aims at proposing an efficient and robust algorithm for the IMB scheme including the application of Gaussian quadrature for computing the solid nodal area, and comparing the IMB with a modified IB method [46]. The remainder of this paper is organised as follows. Section 2 briefly reviews the principles of the lattice Boltzmann method using the single relaxation Bhatnagar-Gross-Krook (BGK) model. The formulation of IMB and an efficient algorithm to identify fluid and solid lattice nodes in IMB are addressed in detail in Section 3, followed by the validation of the proposed method using two benchmark tests in Section 4. Finally, a brief conclusion will be drawn and future work is suggested in Section 5.

2 Lattice Boltzmann Method

The lattice Boltzmann method is a modern numerical approach in Computational Fluid Dynamics. It is originated from the lattice gas automata (LGA) method [48]. In conventional CFD, the fluid phase is treated as continuum governed by the NS equations. The primary variables are pressure, velocity and density. In LBM the fluid domain is divided into regular lattices and the fluid phase is treated as a group of (imaginary) fluid particle packages resided at the lattice nodes. Each particle package includes several particles, such as 9 particles in the commonly used D2Q9 model. The flow of fluid can be achieved through resolving the particle collision and streaming processes, and the Lattice Boltzmann Equation (LBE) is used to solve the streaming and collision process of fluid particles. The primary variables of LBM are the so-called fluid density distribution functions, which are portions of fluid density, associated with the fluid particles. Both mass and momentum of fluid particles are characterised by the fluid density distribution functions.

The Lattice Boltzmann Equation is described by

$$f_i(x+e_i\Delta t, t+\Delta t) - f_i(x,t) = \Omega_i \tag{1}$$

where f_i are the fluid density distribution functions; x and e_i are the coordinate and velocity vectors at the current lattice node; *t* and Ω_i are, respectively, the current time and the collision operator.

There are mainly two models, the single-relaxation-time BGK model [49, 50] and the Multiple-Relaxation-Time (MRT) model [51], for handling the collision process between fluid particles. Because of its better computational efficiency, the BGK model is mostly employed in LBM. In the BGK Model, Ω_i is characterised by a relaxation time τ and the equilibrium distribution functions $f_i^{eq}(x,t)$:

$$\Omega_i = -\frac{\Delta t}{\tau}\left[f_i(x,t) - f_i^{eq}(x,t)\right] + \Delta t F_i \tag{2}$$

$$f_i^{eq} = \omega_i \rho\left(1 + \frac{3}{C^2} e_i \times v + \frac{9}{2C^4}(e_i \times v)^2 - \frac{3}{2C^2} v \times v\right) \tag{3}$$

where *C*, known as the lattice speed, is related to the lattice side, *h*, and the time step, Δt by

$$C = h / \Delta t$$

while ρ and v are the macroscopic fluid density and velocity respectively; F_i is the body force term; and ω_i is a weighting factor.

A popular way of classifying different BGK models is the *DnQm* scheme, where "*Dn*" stands for "*n* dimensions" while "*Qm*" stands for "*m* speeds". In this study, the commonly used *D2Q9* model is adopted for 2D cases. The fluid domain is discretized into square lattices with grid side *h*. Fluid particles at each lattice node are allowed to move to its eight immediate neighbours with different velocities e_i *(i = 1, 2,..,8)*. A proportion of the particles can rest at the node with a zero velocity $\mathbf{e}_0$. The nine discrete velocity vectors in total are defined as

$$e_0 = (0,0)$$

$$e_i = C(\cos\frac{\pi(i-1)}{2}, \sin\frac{\pi(i-1)}{2}) \qquad (i = 1,...,4) \tag{4}$$

$$e_i = C(\cos\frac{\pi(2i-9)}{4}, \sin\frac{\pi(2i-9)}{4}) \qquad (i = 5,...,8)$$

The weighting factors are

$$\omega_0 = \frac{4}{9}, \qquad \omega_{1,2,3,4} = \frac{1}{9}, \qquad \omega_{5,6,7,8} = \frac{1}{36} \tag{5}$$

The macroscopic fluid density ρ and velocity v can be calculated from the distribution functions

$$\rho = \sum_{i=0}^{8} f_i, \qquad \rho v = \sum_{i=1}^{8} f_i e_i \tag{6}$$

The fluid pressure is given by

$$\Delta p = C_S^2 \Delta \rho \tag{7}$$

where C_S is termed the fluid speed of sound and is related to the lattice speed *C*

$$C_S = C/\sqrt{3}$$

3 Immersed Moving Boundary Scheme

3.1 Formulations

The immersed moving boundary scheme was proposed by Noble and Torczynski [34] to overcome fluctuations of the hydrodynamic forces calculated by the modified Bounce Back technique [19]. In this method, a particle is represented by the solid (lattice) nodes which are located within the particle. A solid node is called *interior* if its linked nodes are all solid nodes, while if a solid has at least one link to a fluid node, it is called a *solid boundary* node. A fluid node having at least one link to a solid node is defined as the *fluid boundary* node. Thus, there are four types of node in the IMB scheme: interior solid node, solid boundary node, fluid boundary node and normal fluid node, which are respectively marked in yellow, red, green and blue in an illustrative diagram of IMB in Fig. 1.

Insert Fig. 1

In order to retain the advantages of LBM, namely the locality of the collision operator and the simple linear streaming operator, an additional collision term, Ω_i^s, for the boundary nodes covered partially or fully by the solid is introduced to the standard collision operator of LBM. The modified collision operator for resolving the fluid-solid interaction in IMB is given by

$$\Omega_i = -\frac{\Delta t}{\tau}(1-B)[f_i(x,t) - f_i^{eq}(x,t)] + (1-B)\Delta t F_i + B\Omega_i^S \tag{8}$$

where *B* is a weighting function that depends on the local solid ratio ε, defined as the fraction of the node area (see Fig. 1).

When *B* is zero, the collision operator is reduced to the standard one for fluid. The simplest form of *B* is that it is equal to the local solid ratio ε.

To eliminate Knudsen layer effects (Noble and Torczynski, 1998) appeared in the simplest form, the following form was proposed and a very good accuracy was obtained.

$$B = \frac{\varepsilon(\tau - 0.5)}{(1-\varepsilon) + (\tau - 0.5)} \tag{8a}$$

When $\varepsilon = 0$, *B*=0; $\varepsilon = 1$, *B*=1.

The body force term F_i is determined by the commonly used method proposed by Guo [52]. The additional collision term is based on the bounce-rule for the non-equilibrium part

$$\Omega_i^S = f_{-i}(x,t) - f_i(x,t) + f_i^{eq}(\rho, U_S) - f_{-i}^{eq}(\rho, u) \tag{9}$$

where U_s is the velocity of the solid node (see Fig. 1) and u is the velocity of the fluid at the node.

The velocity of the solid node considering the effect of particle rotation is described by

$$U_S = U_P + \omega \cdot l_P \qquad (l_P = \sqrt{(x-x_P)^2 + (y-y_P)^2}) \tag{10}$$

where U_p and ω are the velocity and angular velocities of the solid particle.

The resultant hydrodynamic force and torque exerted on the solid particle can be calculated by

$$F_f = Ch\left[\sum_n (B_n \sum_i \Omega_i^s e_i) \right] \tag{11}$$

$$T_f = Ch\left\{ \sum_n \left[(x - x_P) \times (B_n \sum_i \Omega_i^s e_i) \right] \right\} \tag{12}$$

Compared to the modified bounce-back rule, this more accurate and smooth lattice representation of a solid particle shape is able to reduce the fluctuation of the computed hydrodynamic forces.

3.2 Computational procedure

Although the IMB scheme has been successfully applied in the coupled DEM-LBM technique, to the best of the authors' knowledge the algorithm of IMB was seldom reported [59, 61]. As mentioned before, there are four types of nodes in the IMB scheme. It is essential to identify the correct type for each node in order to accurately calculate the hydrodynamic forces, particularly those contributed by fluid and solid boundary nodes. Also, as the particles could be continuously moving, this node type identification procedure has to be performed at each time step. Consequently, how to efficiently identify both fluid and solid boundary nodes and compute the solid ratio ε used in Eq. (8a) will have a substantial impact on the overall computational cost of DEM-LBM for a particle-fluid interaction problem, particularly when a large number of particles are involved.

3.2.1 Searching boundary nodes

Insert Fig. 2

Here we propose an efficient boundary tracing procedure (See Fig. 2) for searching solid and fluid boundary nodes of an arbitrarily shaped convex particle. It is derived mainly based on the modification of the classic Bresenham's line algorithm [54] for plotting a line segment on a computer screen.

Insert Fig. 3

The essence of our algorithm is linearly searching solid and fluid boundary nodes by moving the red and dotted mesh shown in Figure 2. The movement of the dotted mesh is controlled by the directional pointer MARCH(:,:) (see Table 1). To make the search method general and efficient, the involved domain is divided into four zones based on the centroid of a convex solid particle, and the searching method is applicable to four different zones. Only the searching direction is updated when the dotted mesh moves to a new zone according to the directional pointer.

The tracing procedure is illustrated in Fig. 3 for a circular particle as an example. It should be mentioned that this search method is also applicable to convex polygons or any convex shape. The detailed implementation of the proposed algorithm is as follows:

Algorithm of IMB

DO IP=1, NP (Loop over solid particles)

Step 1 Divide the solid particle into four zones with the geometric centre as the local origin of coordinates (see Fig. 2).
Locate the first (lowest) solid boundary node A and set the zone number (IZONE) as 4, then record the coordinates of four nodes (A, B, C and D) of the mesh marked by red dotted lines.

Step 2 **IF** (B is within the solid particle) **THEN**
(like case 2 or case 3 in Fig. 4)
    mark the status of four nodes and get the values of MARCH(X, IZONE) and MARCH(Y, IZONE);
ELSE
(like case 1 or case 4)
    mark the status of four nodes and get the values of MARCH(X, IZONE-1) and MARCH(Y, IZONE-1)
END IF

Step 3 Move to the next mesh according to MARCH and update the positions of the corresponding nodes A-D.

Step 4 **IF** (node A reach the first (lowest) solid boundary node) **THEN**
    GOTO step 5
ELSE
    IF (node A reach new zone) **THEN**

```
                        update IZONE=IZONE-1
                        clockwise rotate this mesh by 90 degrees so that A replaces D.
                        update positions of nodes A-D.
                ELSE
                        GOTO step 2
                ENDIF
            END IF
Step 5      IP=IP+1

END DO
```

Insert Table 1

Insert Fig. 4

It is not difficult to deduce that the computational complexity of the above boundary tracing algorithm is linear: *O(n)*.

$$O(n) = \pi \cdot d/h \tag{13}$$

where *d* is diameter of the particle; *n=d/h*.

In our previous boundary tracing method [59], the fluid boundary and solid boundary nodes are separately searched and the computational complexity is

$$O(n) = 2 \cdot \pi \cdot d/h \tag{14}$$

Compared to our previous algorithm, the identification cost can be almost halved.

3.2.1 Computing nodal solid ratio

After both fluid and solid boundary nodes are identified, the solid ratio associated with each of these nodes will be calculated. Calculation of the solid ratio is a mathematical integration problem and it takes significant computational cost. The geometrical approximation is commonly used to estimate the solid ratio. Two geometric approximation methods are discussed in reference [38]. The accuracy of these methods is dependent on the (sub) mesh size.

In this paper, an efficient Gaussian quadrature is used to calculate the nodal solid area. Take the general case shown in Fig. 5 for example. The Gaussian integration of order N for the standard interval [*a=-1, b=1*] is shown by

Insert Fig. 5

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{N} w_i f(\xi_i) \tag{15}$$

where ξ_i and w_i are the Gaussian positions and weights respectively; and N is the number of Gaussian points.

To facilitate the computing, the Gaussian quadrature for general intervals [*a, b*] is given by

$$\int_a^b f(x)dx \approx \frac{b-a}{2}\sum_{i=1}^{N} w_i f\left(a+\frac{b-a}{2}(1+\xi_i)\right) \tag{16}$$

The computation of the solid ratios of both fluid and solid boundary nodes can be divided into four cases (shown in Fig. 6). In Case 3, the solid area includes two parts. The first rectangular area can be easily calculated, and the second part can be obtained by Gaussian quadrature (Eq. 16) between points *a* and *b*. Cases 1, 2 & 4 can be directly integrated by Eq. 16 between *a* and *b*.

Insert Fig. 6

In order to examine the accuracy of Gaussian quadrature, a special case (see Fig. 7) where the nodal area is covered by a quarter of circle is selected so that the solid area computed can be compared with the analytical solution. The numbers of Gaussian points (2 to 5) are respectively adopted and the corresponding numerical errors are given in Fig. 8. It can be found that the Gaussian quadrature has very high accuracy. The computed solid area using two Gaussian points only has 4.0% error. While, the Monte Carlo method using 500 points can only achieve 4.5% error, and it is reported to be an efficient method for computing the solid ratio [60, 61]. Hence, the Gaussian quadrature is much more efficient. Quantitative validations of the proposed IMB algorithm will be given in Section 4.

It should be highlighted that the Gaussian quadrature can be straightforwardly used for computing the solid ratio in 3-dimensional (3D) problems where the computing cost for the solid ratio is much more expansive than that in 2D simulation. Numerical errors of Gaussian quadrature and Monte Carlo in a certain 3D simulation are given in Fig. 9. It is found that to achieve 4.0% error Gaussian quadrature only needs 9 Gaussian points; while, the Monte Carlo method needs at least 1000 points.

Finally, the collision and hydrodynamic force/toque calculations for these nodes can be calculated using Eqs. 11 and 12.

Insert Fig. 7

Insert Fig. 8

Insert Fig. 9

4 Validations and discussions

In order to examine the feasibility and accuracy of the proposed IMB procedure, the standard benchmark, flow past a cylinder, will be first carried out. Then, the comparison of the IMB scheme and the implicit velocity correction based IBM-LBM [46] is made through the extensively investigated single particle sedimentation in a viscous fluid.

4.1 Flow past a cylinder

Flow past a cylinder has long been a subject of interest to researchers in fluid dynamics. Extensive work including experiments and numerical simulations has been undertaken. In our simulation, this problem concerns steady and unsteady flows around a circular cylinder placed in a long rectangular channel. The channel (see Fig. 10) is 1 cm in height (the Y direction) and 8 cm in length (the X direction). A cylinder of 0.2 cm in diameter is placed at the position (2.0 cm, 0.5 cm). Both top and bottom boundaries are solid walls where the no-slip boundary condition is applied. The pressure boundary condition is applied at both the inlet (left boundary) and the outlet (right boundary) of the fluid domain (the pressure difference is 7.5 kPa). The fluid domain is divided into 800×100 lattices with spacing h = 0.01 cm. The relaxation parameter τ is 0.5001.

When the fluid approaches the front side of the cylinder, the fluid pressure increases and the fluid is forced to move along the cylinder surface. With the Reynolds number exceeding a certain value, the fluid cannot follow the cylinder surface to the rear side but separates from both sides and a pair of symmetric vortices are formed in the near wake (at t = 0.6667 s). As the Reynolds number (Re>45) increases further, the wake becomes unstable. One vortex will draw the opposite vortex across the wake, and then vortex shedding is initiated at t = 2.2667 s where the Reynolds number further increases to about 100.

The streamline and corresponding velocity contours at different time instants are displayed in Fig. 10. The quantitative comparison of the drag coefficient calculated using the proposed LBM procedure with the experimental, theoretical and CFD numerical results [55] is made in Fig. 11. It is found that the drag coefficients for Reynold numbers (Re) between 10 and 110 match the experimental and CFD data very well; while there are certain differences when Re is lower than 10. Interestingly, for the Stokes flow (Re<1) the proposed LBM procedure is much closer to the theoretical result governed by Eq. 17.

$$C_d = \frac{24}{\mathrm{Re}} \tag{17}$$

To examine the convergence of the algorithm in terms of the ratio of the particle diameter *d* to the grid size *h*, four fluid cell resolutions with different grid sizes 0.04, 0.02, 0.01 and 0.005 cm are simulated and the corresponding errors of the drag coefficient are presented in Fig. 12. In the convergence test, the Reynolds is around 104. The reference value of the drag coefficient is select as 1.10 from the experimental results in Fig. 11. It can be found that the drag coefficient is sensitive to the grid size or the size ratio *d/h*: the error of the drag coefficient decreases with the increase of the grid size ratio, but converges when the size ratio is greater than 20, where the error is lower than 3%.

Insert Fig. 10

Insert Fig. 11

Insert Fig. 12

4.2 Single particle sedimentation in viscous fluid

The second problem has also been extensively employed to validate numerical methods proposed for resolving fluid-particle interactions. In our simulation, a water-filled tube with 2 cm in width (the X-direction) and 6 cm in height (the Y-direction) is used. The fluid domain is divided into 200×600 square lattices with spacing h=0.1 mm. The kinematic viscosity and density of fluid are 1.0×10^{-6} m^2/s and 1000 kg/m^3, respectively. The density of the solid particle is 3000 kg/m^3, and its radius is 0.125 cm. The four boundaries of the fluid domain are treated as stationary walls and thus the no-slip boundary condition is imposed. Initially, the particle is positioned at point (1cm, 4cm) in rest. Due to the gravity, the particle will fall down. The IMB scheme is employed to resolve the particle-fluid interaction. The sedimentation process and fluid velocity contours of fluid at five different time instants, 0.093s, 0.14s, 0.187s, 0.21s, 0.233s, are depicted in Fig. 13. When the particle is released from the rest, it falls down in an accelerated motion due to the gravitational force. After a while it will move with a constant or termination speed as the gravity, hydrodynamic and buoyancy forces acting on the particle reach an equilibrium state.

To valid the proposed algorithm of IMB, the same simulation using the implicit velocity correction based IBM scheme [46] is performed. The variation of particle velocity and position in the vertical direction and the hydrodynamic forces applied to the particle with respect to time are compared in Figs. 14-16. The computing costs for this simulation using IMB proposed in this paper, our previous IMB algorithm and IBM are 1h 53m 47s, 3h 11m 13s and 3h 26m 56s, respectively, on the computer (Intel Core i5-3450 CPU@3.10GHz). It can be seen, the proposed IMB algorithm can reduce at least 1/3 of the computational cost of IBM and the previous IMB algorithm.

Figs. 14-16 show that the motions of the particle simulated by both IMB and IBM schemes match very well at the early stage of sedimentation and some small discrepancies appear later. However, it is very noticeable that the hydrodynamic force calculated by the IMB scheme evolves smoothly except at one point where the particle collides the bottom boundary; while the drag forces obtained from the IBM scheme fluctuate heavily around those values calculated by IMB, indicating that the IBM scheme has an underlying issue which may adversely affect its numerical stability when a large number of particles are present and/or for more complex fluid flow cases.

Insert Fig. 13

In order to investigate the influence of grid sizes on the hydrodynamic forces calculated by IMB, we change the diameter of the tube and the radius of the particle used in the above simulation to 3 cm and 0.15 cm, respectively. Therefore, the grid sizes (h) of 0.0625, 0.1, 0.15, 0.3 and 0.6 mm are chosen so that the ratios of the particle diameter to the grid size become 48, 30, 20, 10 and 5. From Fig. 17 it can be found that the coarse grids result in larger drag forces than the fine grids, but with the decrease of the grid size, a convergence of the drag force applied to the particle can be observed. When the ratio of particle diameter to grid size reaches 20 or more, a reasonably accurate simulation of fluid-particle interactions can be achieved.

From the aforementioned numerical tests, the accuracy and feasibility of the proposed algorithm have been well demonstrated. The simulation of the fluid-particle interaction is accurate and stable due to the smooth representation of the particle surface. In addition, the applicability of IMB to the fluid-particle systems involving a large number of particles has also been demonstrated in our previous work [26, 56, 57]. Compared to the FEM and FVM, this procedure can be efficiently applied to fluid-multiple particle systems with minimum assumption, and no adaptive meshes are needed for considering moving particles.

Insert Fig. 14

Insert Fig. 15

Insert Fig. 16

Insert Fig. 17

5 Conclusions

In this paper, an efficient and robust IMB algorithm has been developed to efficiently identify both fluid and solid boundary nodes associated with a solid particle, and compute the solid nodal ratio. This novel algorithm is demonstrated through two benchmark tests. Some conclusions can be drawn.

1) The novel boundary tracing procedure proposed in this paper is robust and efficient. Compared to our previous linear searching algorithm, the computing cost is reduced by half. Although only circular particles are used to illustrate the novel searching method, the algorithm itself is universal and can be easily implemented for convex polygons and other shapes. The only difference is that the equation of circle will be replaced by the polygon equation or the expression for the convex shape concerned while determining the type of boundary nodes.

2) The Gaussian quadrature for computing the solid nodal ratio is of high accuracy and efficiency. Compare to another commonly used method, Monte Carlo Simulation, to achieve the same and high accuracy only a few points are needed, while the latter needs at least a thousand. The importance of Gaussian quadrature will be more apparent in 3-dimensional simulations where the computation of the solid ratio is more complicated and time-consuming.

3) The accuracy of the proposed procedure for fluid-particle interactions is dependent on the ratio of the particle size to the grid size of LBM, and it has been proven that to attain a reasonable result, the ratio should be at least 20 for 2D cases.

4) The IMB scheme is more stable than the IBM scheme in terms of the calculated hydrodynamic force, and therefore is more robust to simulate problems with a large number of particles and/or more complex fluid flow patterns.

However, the proposed boundary tracing algorithm is only valid for the 2-dimensional fluid-particle coupling. Different tracing approaches have to be used for 3-dimensional problems which will be reported elsewhere.

References

1. Ye, T., Mittal, R., Udaykumar, H. S. & Shyy, W. An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries. *Journal of Computational Physics* 1999; 156: 209-240.

2. Udaykumar, H. S., Mittal, R. & Rampunggoon, P. Interface tracking finite volume method for complex solid–fluid interactions on fixed meshes. *Communications in Numerical Methods in Engineering* 2002;18**:** 89-97.
3. Mittal, R. & Iaccarino, G. Immersed boundary methods. *Annu. Rev. Fluid Mech.* 2005; 37**:** 239-261.
4. Feng, J., Hu, H. H. & Joseph, D. D. Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 1. Sedimentation. *Journal of Fluid Mechanics* 1994; 261**:** 95-134.
5. Hu, H. H. Direct simulation of flows of solid-liquid mixtures. *International Journal of Multiphase Flow* 1996; 22**:** 335-352.
6. Johnson, A. A. & Tezduyar, T. E. Simulation of multiple spheres falling in a liquid-filled tube. *Computer Methods in Applied Mechanics and Engineering* 1996; 134**:** 351-373.
7. Maury, B. Direct Simulations of 2D Fluid-Particle Flows in Biperiodic Domains. *Journal of Computational Physics* 1999; 156**:** 325-351.
8. Tsuji, Y., Kawaguchi, T. & Tanaka, T. Discrete particle simulation of two-dimensional fluidized bed. *Powder Technology* 1993; 77**:** 79-87.
9. Xu, B. H. & Yu, A. B. Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics. *Chemical Engineering Science* 1997; 52**:** 2785-2809.
10. Kafui, K. D., Thornton, C. & Adams, M. J. Discrete particle-continuum fluid modelling of gas–solid fluidised beds. *Chemical Engineering Science* 2002; 57**:** 2395-2410.
11. Yu, A. B. & Xu, B. A. Particle-scale modelling of gas-solid flow in fluidisation. *Journal of Chemical Technology & Biotechnology* 2003; 78**:** 111-121.
12. Zhu, H. P., Zhou, Z. Y., Yang, R. Y. & Yu, A. B. Discrete particle simulation of particulate systems: Theoretical developments. *Chemical Engineering Science* 2007; 62**:** 3378-3396.
13. Wu, C.-Y. & Guo, Y. Numerical modelling of suction filling using DEM/CFD. *Chemical Engineering Science* 2012; 73**:** 231-238.
14. Höfler, K. & Schwarzer, S. Navier-Stokes simulation with constraint forces: Finite-difference method for particle-laden flows and complex geometries. *Physical Review E* 2000; 61**:** 7146.
15. Zhang, Z. & Prosperetti, A. A method for particle simulation. *Journal of applied mechanics* 2003; 70**:** 64-74.
16. Perrin, A. & Hu, H. H. An Explicit Finite-Difference Scheme for Simulation of Moving Particles. *In:* BALACHANDAR, S. & PROSPERETTI, A. (eds.) *IUTAM Symposium on Computational Approaches to Multiphase Flow* 2006; Springer Netherlands.
17. Peskin, C. S. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; 25**:** 220-252.
18. Peskin, C. S. The immersed boundary method. *Acta Numerica* 2002; 11**:** 479-517.
19. Ladd, A. J. C. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *Journal of Fluid Mechanics* 1994; 271**:** 285-309.

20. Zhang, J., Fan, L.-S., Zhu, C., Pfeffer, R. & Qi, D. Dynamic behavior of collision of elastic spheres in viscous fluids. *Powder Technology* 1999; 106**:** 98-109.
21. Ladd, A. J. C. & Verberg, R. Lattice-Boltzmann Simulations of Particle-Fluid Suspensions. *Journal of Statistical Physics* 2001; 104**:** 1191-1251.
22. Mansouri, M., Delenne, J. Y., El Youssoufi, M. S. & Seridi, A. A 3D DEM-LBM approach for the assessment of the quick condition for sands. *Comptes Rendus Mécanique* 2009; 337**:** 675-681.
23. Delenne, J.-Y., Mansouri, M., Radjaï, F., Youssoufi, M. S. E. & Seridi, A. Onset of Immersed Granular Avalanches by DEM-LBM Approach. . *In:* BONELLI, S., DASCALU, C. & NICOT, F. (eds.) *Advances in Bifurcation and Degradation in Geomaterials* 2011. Springer Netherlands.
24. Lominé, F., Scholtès, L., Sibille, L. & Poullain, P. Modeling of fluid–solid interaction in granular media with coupled lattice Boltzmann/discrete element methods: application to piping erosion. *International Journal for Numerical and Analytical Methods in Geomechanics* 2013; 37**:** 577-596.
25. Feng, Z. G. & Michaelides, E. E. The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. *Journal of Computational Physics* 2004; 195**:** 602-628.
26. Feng, Y. T., Han, K. & Owen, D. R. J. Coupled lattice Boltzmann method and discrete element modelling of particle transport in turbulent fluid flows: Computational issues. *International Journal for Numerical Methods in Engineering* 2007; 72**:** 1111-1134.
27. Han, K., Feng, Y. T. & Owen, D. R. J. Coupled lattice Boltzmann and discrete element modelling of fluid-particle interaction problems. *Computers and Structures* 2007; 85**:** 1080-1088.
28. Filippova, O. & Hänel, D. Grid Refinement for Lattice-BGK Models. *Journal of Computational Physics* 1998; 147**:** 219-228.
29. Mei, R., Luo, L.-S. & Shyy, W. An Accurate Curved Boundary Treatment in the Lattice Boltzmann Method. *Journal of Computational Physics* 1999;155**:** 307-330.
30. Mei, R., Yu, D., Shyy, W. & Luo, L.-S. Force evaluation in the lattice Boltzmann method involving curved geometry. *Physical Review E* 2002; 65**:** 041203.
31. Lallemand, P. & Luo, L.-S. Lattice Boltzmann method for moving boundaries. *Journal of Computational Physics* 2003;184**:** 406-421.
32. Yu, D., Mei, R. & Shyy, W. A unified boundary treatment in lattice Boltzmann method. *New York: AIAA* 2003; 953.
33. Kao, P. H. & Yang, R. J. An investigation into curved and moving boundary treatments in the lattice Boltzmann method. *Journal of Computational Physics* 2008; 227**:** 5671-5690.
34. Noble, D. R. & Torczynski, J. R. A Lattice-Boltzmann Method for Partially Saturated Computational Cells. *International Journal of Modern Physics C* 1998; 09**:** 1189-1201.
35. Cook, B. K., Noble, D. R. & Williams, J. R. A direct simulation method for particle-fluid systems. *Engineering Computations* 2004; 21**:** 151-168.
36. Strack, O. E. & Cook, B. K. Three-dimensional immersed boundary conditions for moving solids in the lattice-Boltzmann method. *International Journal for Numerical Methods in Fluids* 2007; 55**:** 103-125.

37. Feng, Y. T., Han, K. & Owen, D. R. J. Combined three-dimensional lattice Boltzmann method and discrete element method for modelling fluid-particle interactions with experimental assessment. *International Journal for Numerical Methods in Engineering* 2010; 81**:** 229-245.

38. Owen, D. R. J., Leonardi, C. R. & Feng, Y. T. An efficient framework for fluid–structure interaction using the lattice Boltzmann method and immersed moving boundaries. *International Journal for Numerical Methods in Engineering* 2011, 87**:** 66-95.

39. Han, Y. & Cundall, P. A. LBM–DEM modeling of fluid–solid interaction in porous media. *International Journal for Numerical and Analytical Methods in Geomechanics* 2012; 37:1391–1407.

40. Cui, X., Li, J., Chan, A. & Chapman, D. Coupled DEM–LBM simulation of internal fluidisation induced by a leaking pipe. *Powder Technology* 2014; 254**:** 299-306.

41. Peng, Y., Shu, C., Chew, Y. T., Niu, X. D. & Lu, X. Y. Application of multi-block approach in the immersed boundary–lattice Boltzmann method for viscous fluid flows. *Journal of Computational Physics* 2006; 218**:** 460-478.

42. Shu, C., Liu, N. & Chew, Y.T. A novel immersed boundary velocity correction–lattice Boltzmann method and its application to simulate flow past a circular cylinder. *Journal of Computational Physics* 2007; 226**:** 1607-1622.

43. Dupuis, A., Chatelain, P. & Koumoutsakos, P. An immersed boundary–lattice-Boltzmann method for the simulation of the flow past an impulsively started cylinder. *Journal of Computational Physics* 2008; 227**:** 4486-4498.

44. Wu, J. & Shu, C. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *Journal of Computational Physics* 2009; 228**:** 1963-1979.

45. Wu, J. & Shu, C. Particulate flow simulation via a boundary condition-enforced immersed boundary-lattice Boltzmann scheme. *Communications in Computational Physics* 2010; 7**:** 793-812.

46. Dash, S. M., Lee, T. S., Lim, T. T. & Huang, H. A flexible forcing three dimension IB–LBM scheme for flow past stationary and moving spheres. *Computers & Fluids* 2014; 95**:** 159-170.

47. Dash, S. M., Lee, T. S. & Huang, H. Particle Sedimentation in a Constricted Passage Using a Flexible Forcing IB-LBM Scheme. *International Journal of Computational Methods* 2015;12**:** 1350095.

48. Mcnamara, G. R. & Zanetti, G. Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. *Physical Review Letters* 1988; 61**:** 2332-2335.

49. Chen, S., Chen, H. Martinez, D. O. & Matthaeus W. H. Lattice Boltzmann model for simulation of magnetohydrodynamics. *Phys. Rev. Lett.* 1991; 67**:** 3776-3779.

50. Chen, H., Chen S. & Matthaeus, W. H. Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. *Phys, Rev. A* 1992; 45**:** 5339-5342.

51. D'humieres, D., Ginzburg, I., Krafczyk, M., Lallemand, P. & Luo, L. S. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 2002; 360**:** 437-451.

52. Guo, Z., Zheng, C. & Shi, B. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E* 2002; 65**:** 046308.
53. Han, Y. and Cundall, P. A. 2011. Resolution sensitivity of momentum-exchange and immersed boundary methods for solid–fluid interaction in the lattice Boltzmann method. Int. J. Numer. Meth. Fluids, 67: 314–327.
54. Bresenham J.E. Algorithm for computer control of a digital plotter. IBM Systems Journal 1965, 4: 25-30.
55. Sato, M. and Kobayashi T., A Fundamental study of the flow past a circular cylinder using Abaqus/CFD", *2012 SIMULIA Community Conference*. Providence RI USA.
56. Wang M, Feng Y T, Wang C Y. Numerical investigation of initiation and propagation of hydraulic fracture using the coupled Bonded Particle–Lattice Boltzmann Method. *Computers & Structures* 2017, 181: 32-40.
57. Wang, M., Feng, Y.T. & Wang, C.Y. Coupled bonded particle and lattice Boltzmann method for modelling fluid–solid interaction, *International Journal of Numerical and Analytical Methods in Geomechanics* 2016, 40: 1383-1401.
58. Rettinger, C., & Rüde, U. A comparative study of fluid-particle coupling methods for fully resolved lattice Boltzmann simulations, *Computers & Fluids* 2017, 154: 74-89.
59. Wang, M., Feng, Y. T., Wang, Y., & Zhao, T. T. Periodic boundary conditions of discrete element method-lattice Boltzmann method for fluid-particle coupling. *Granular Matter*, 2017,19(3), 43.
60. Jones, B. D. & Williams, J. R. Fast computation of accurate sphere-cube intersection volume. *Engineering Computations* 2017, 34(4), 1204-1216.
61. Wang, M., Feng, Y.T., Pande, G.N., Zhao, T.T. A coupled 3-dimensional bonded discrete element and lattice Boltzmann method for fluid-solid coupling in cohesive geomaterials. *Int J Numer Anal Methods Geomech* 2018, 42(12), 1405-1424.
62. Wang, D., Leonardi, C.R. and Aminossadati, S.M. Improved coupling of time integration and hydrodynamic interaction in particle suspensions using the lattice Boltzmann and discrete element methods. *Computers & Mathematics with Applications* 2018, 75(7), 2593-2606.
63. E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, Journal of Computational Physics 161 (2000) 35–60.
64. Ji, C., Munjiza, A. and Williams, J.J.R., 2012. A novel iterative direct-forcing immersed boundary method and its finite volume applications. Journal of Computational Physics, 231(4), 1797-1821.

Figure list

Fig. 1 IMB scheme and definition of local solid ratio e

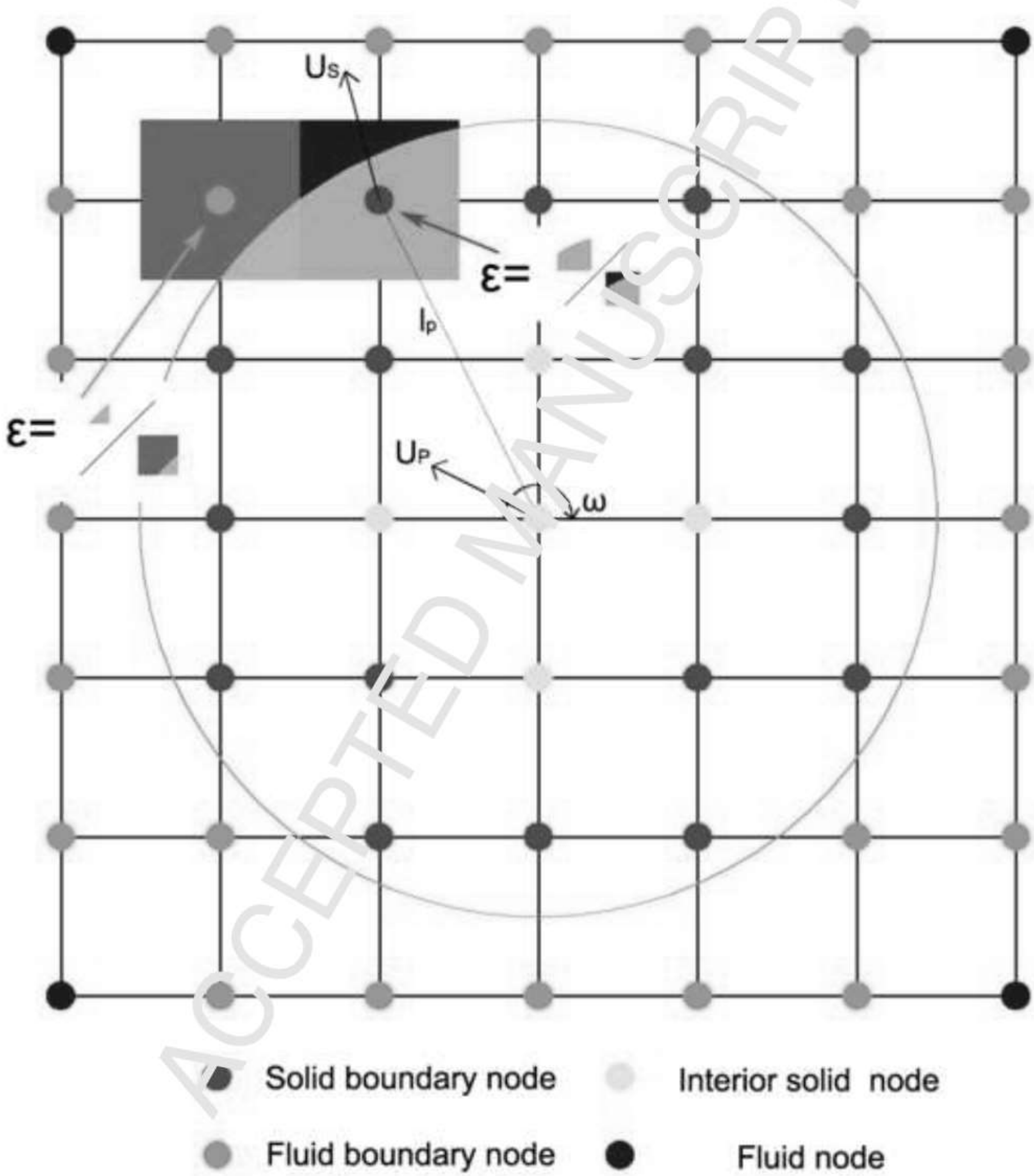


Fig. 2 Boundary nodes tracing scheme
Click here to download high resolution image

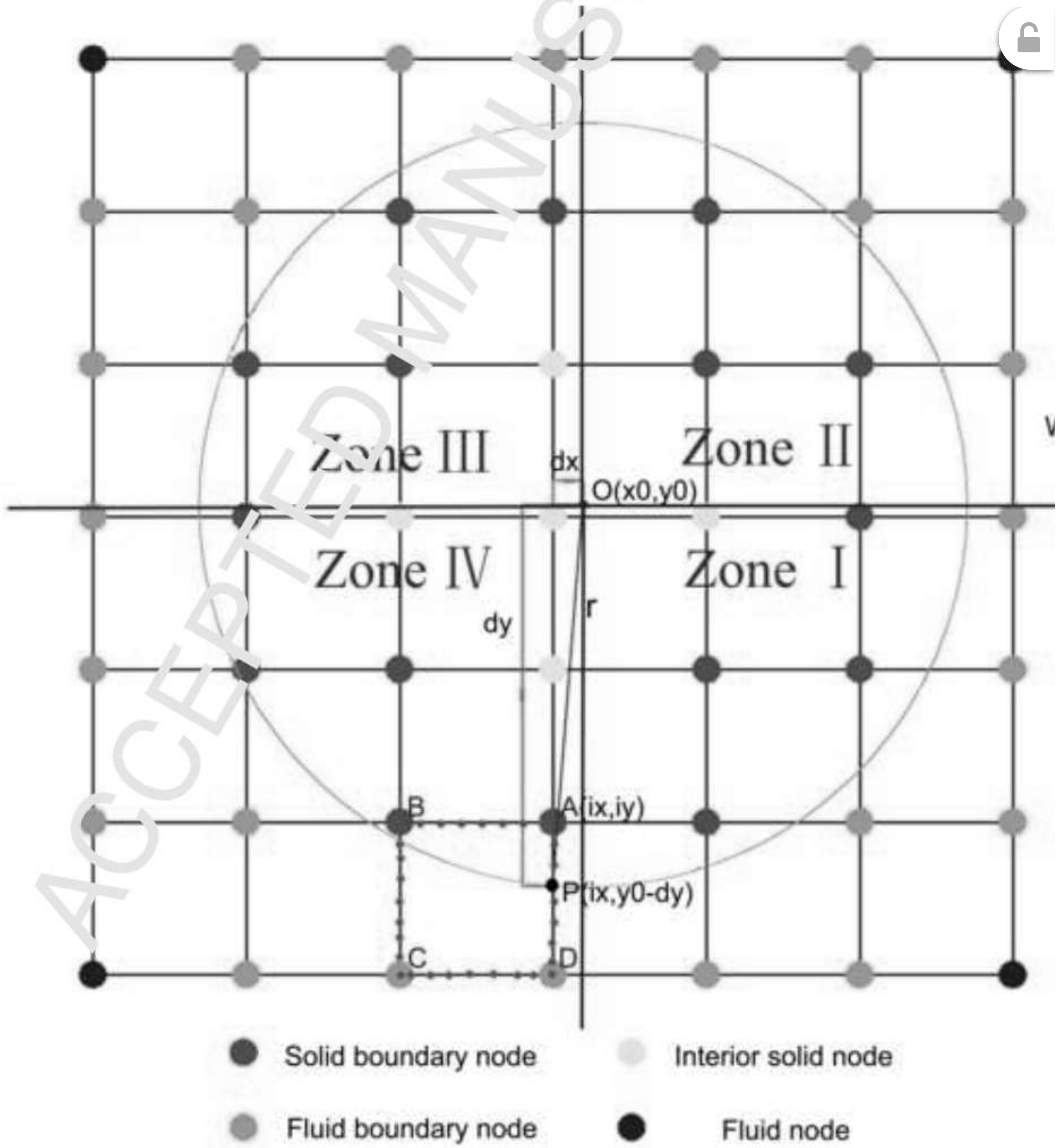


Fig. 3 Flow chart of the searching algorithm

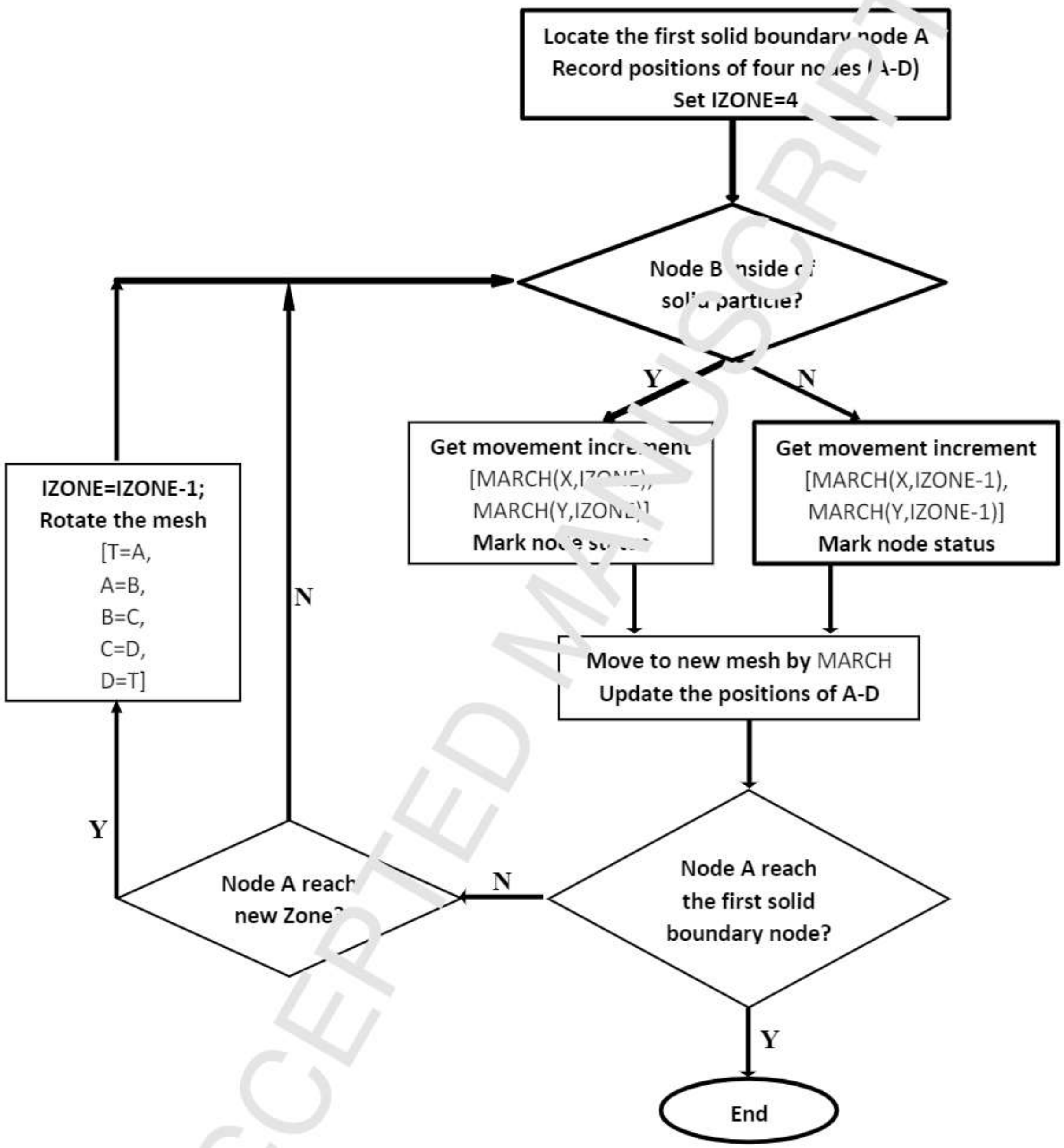


Fig. 4 Potential status of nodes
Click here to download high resolution image

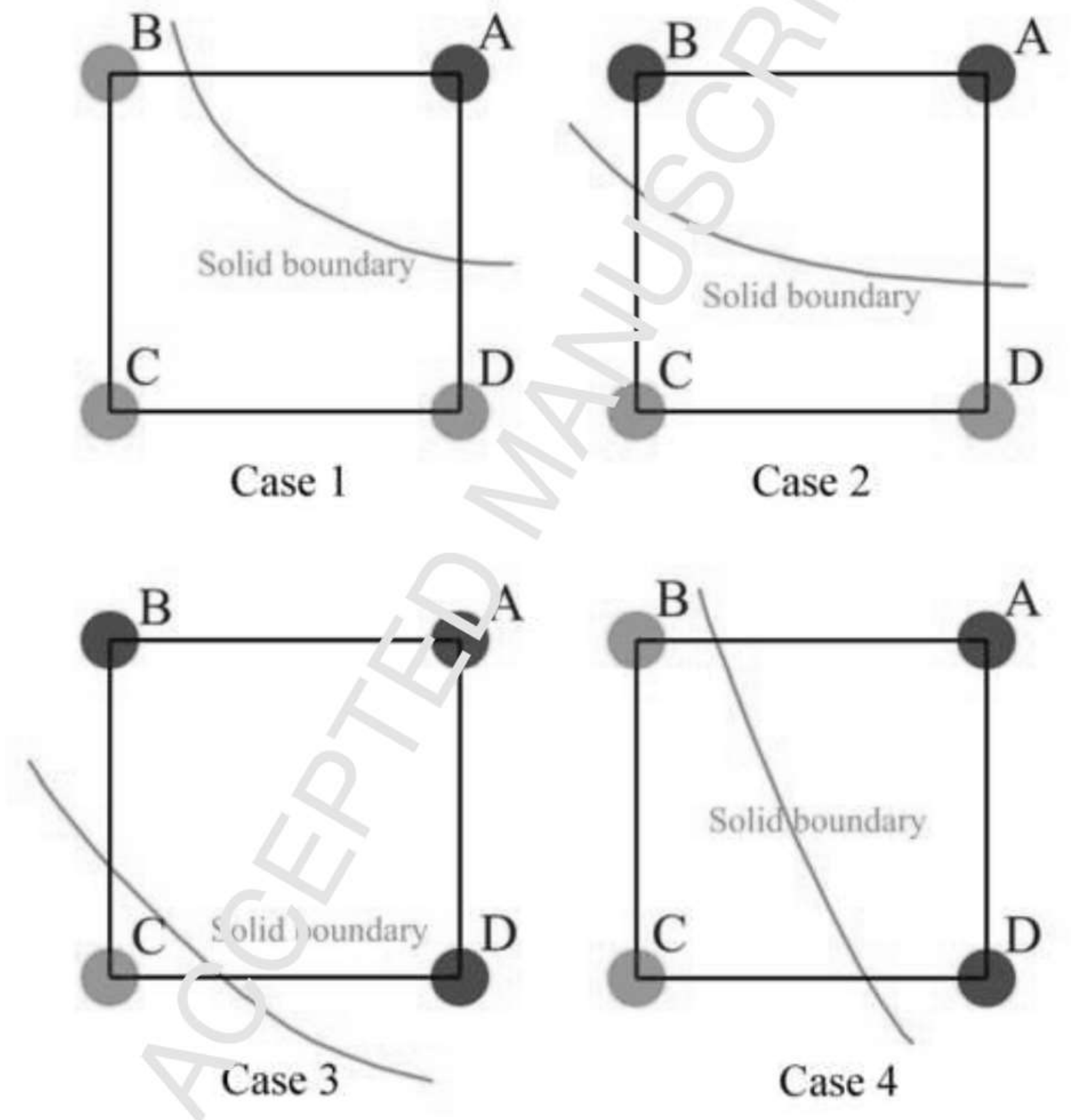


Fig. 5 Gaussian integration for solid nodal area

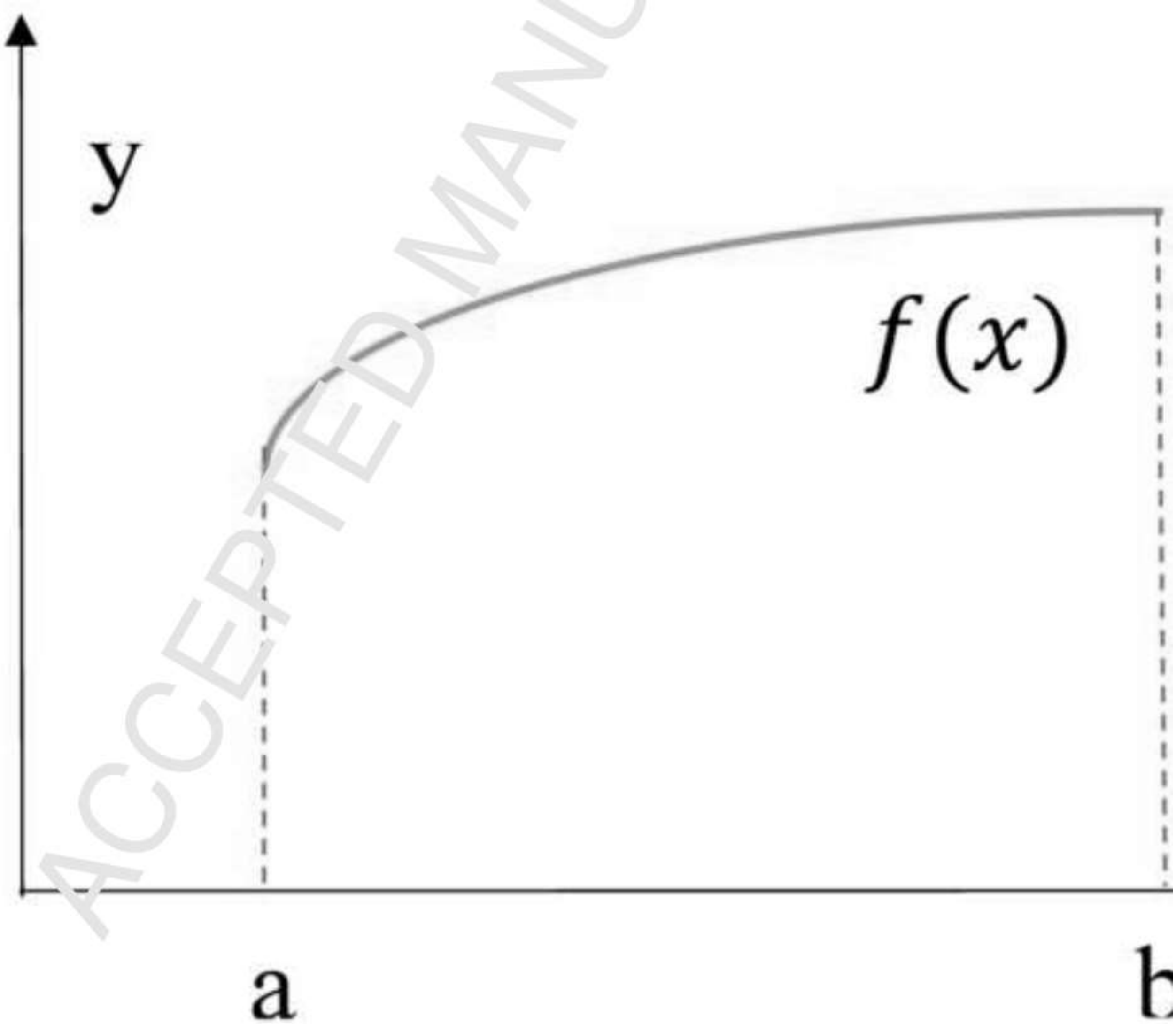


Fig. 6 Four cases in the computation of solid ratio
Click here to download high resolution image

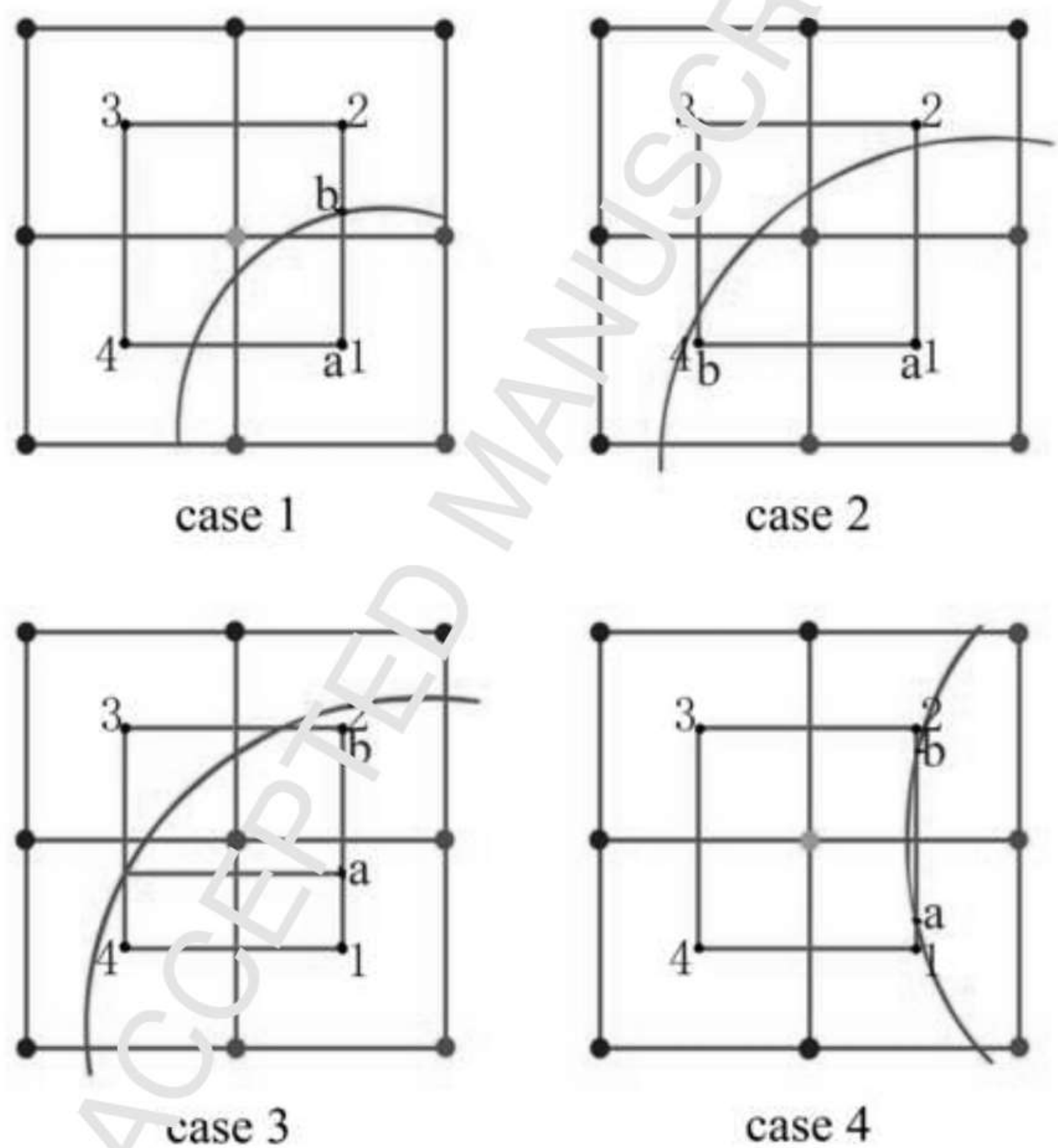


Fig. 7 A special case of solid nodal area with analytical soluti

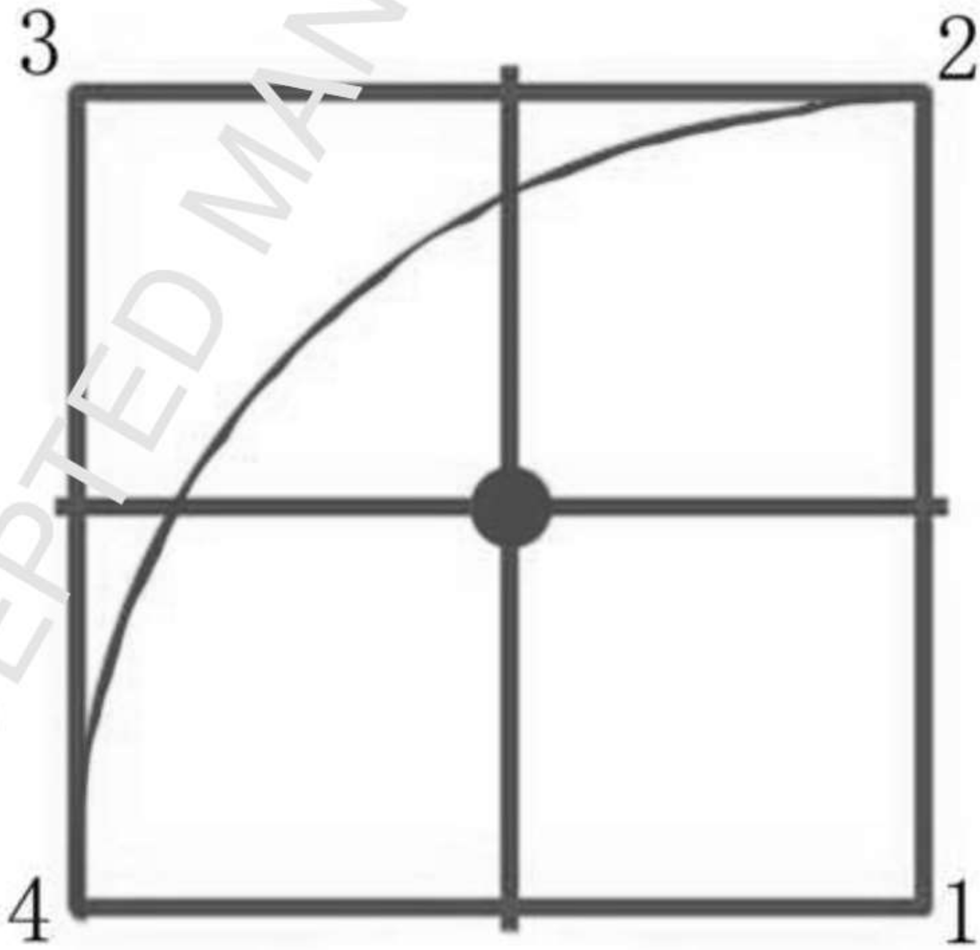
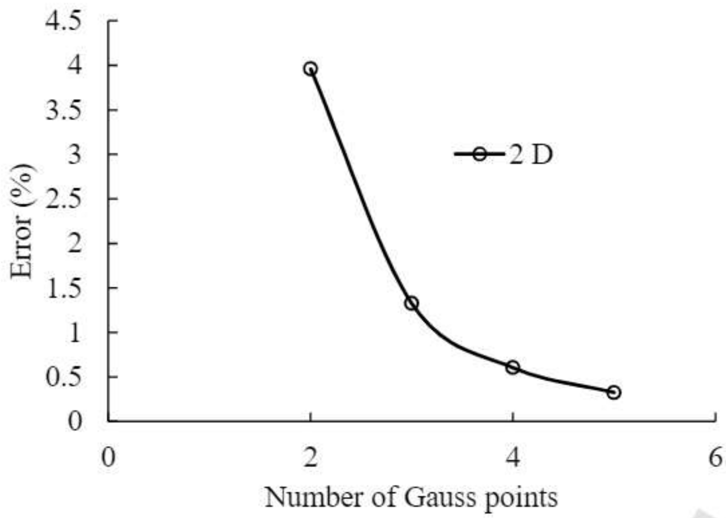
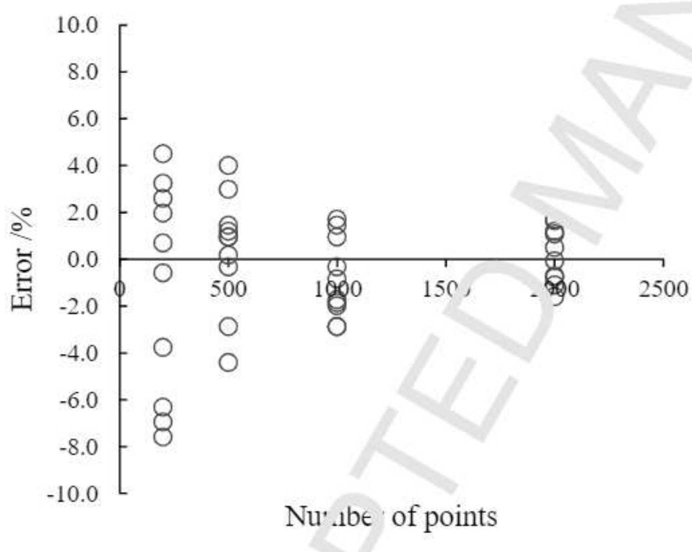


Fig. 8 Numerical errors for 2D simulation

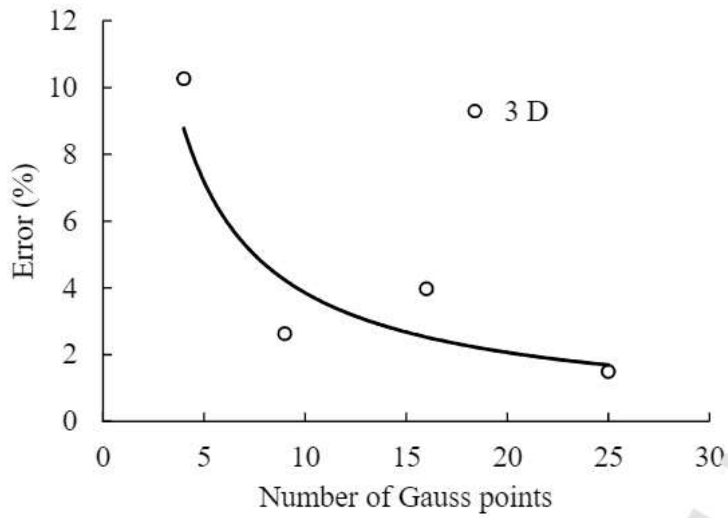


a) Gaussian quadrature

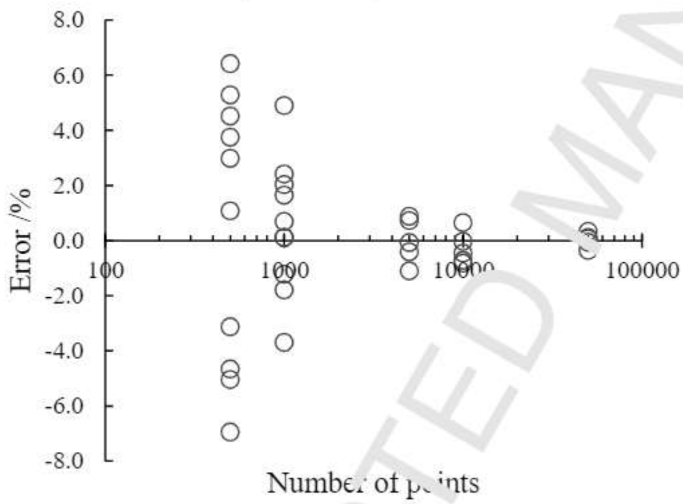


b) Monte Carlo

Fig. 9 Numerical errors for 3D simulation



a) Gaussian quadrature



b) Monte Carlo

Fig. 10 Velocity contours at different stages
Click here to download high resolution image

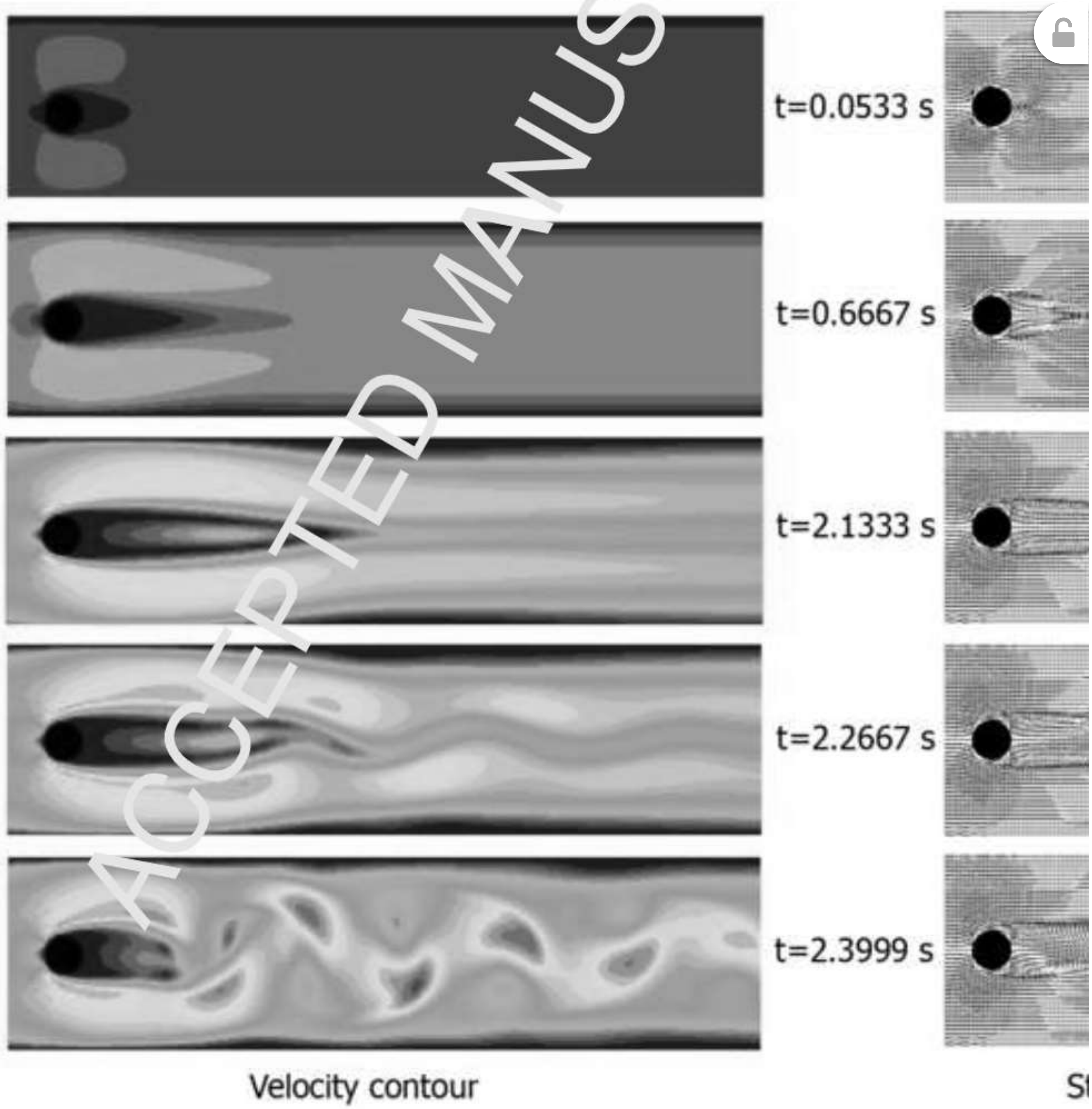


Fig. 11 Variation of drag coefficient with Reynold number

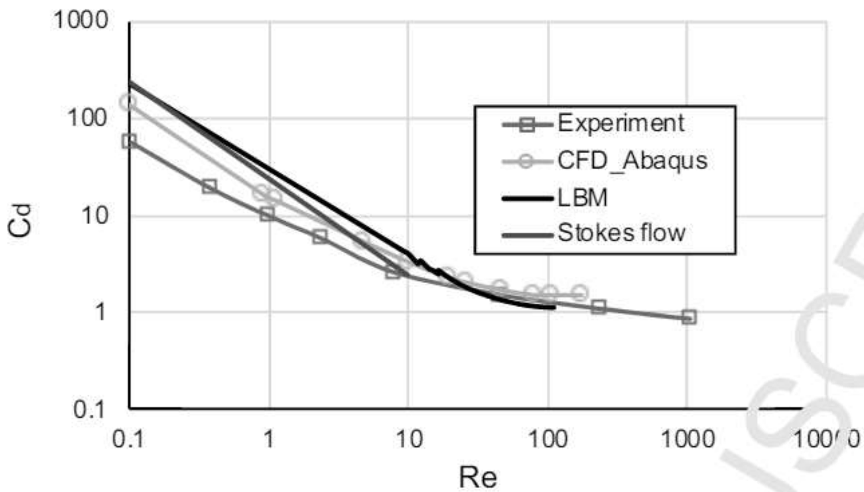


Fig. 12 Grid size effect on the error of drag coefficient

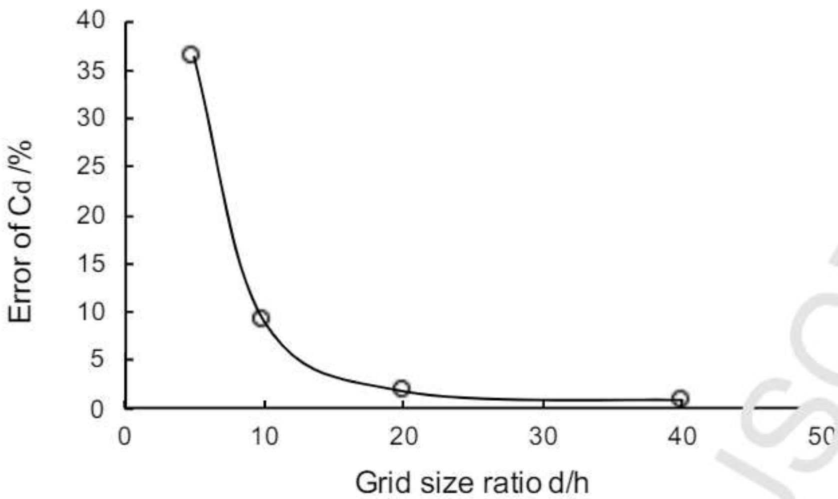


Fig. 13 Total velocity contour at different stages

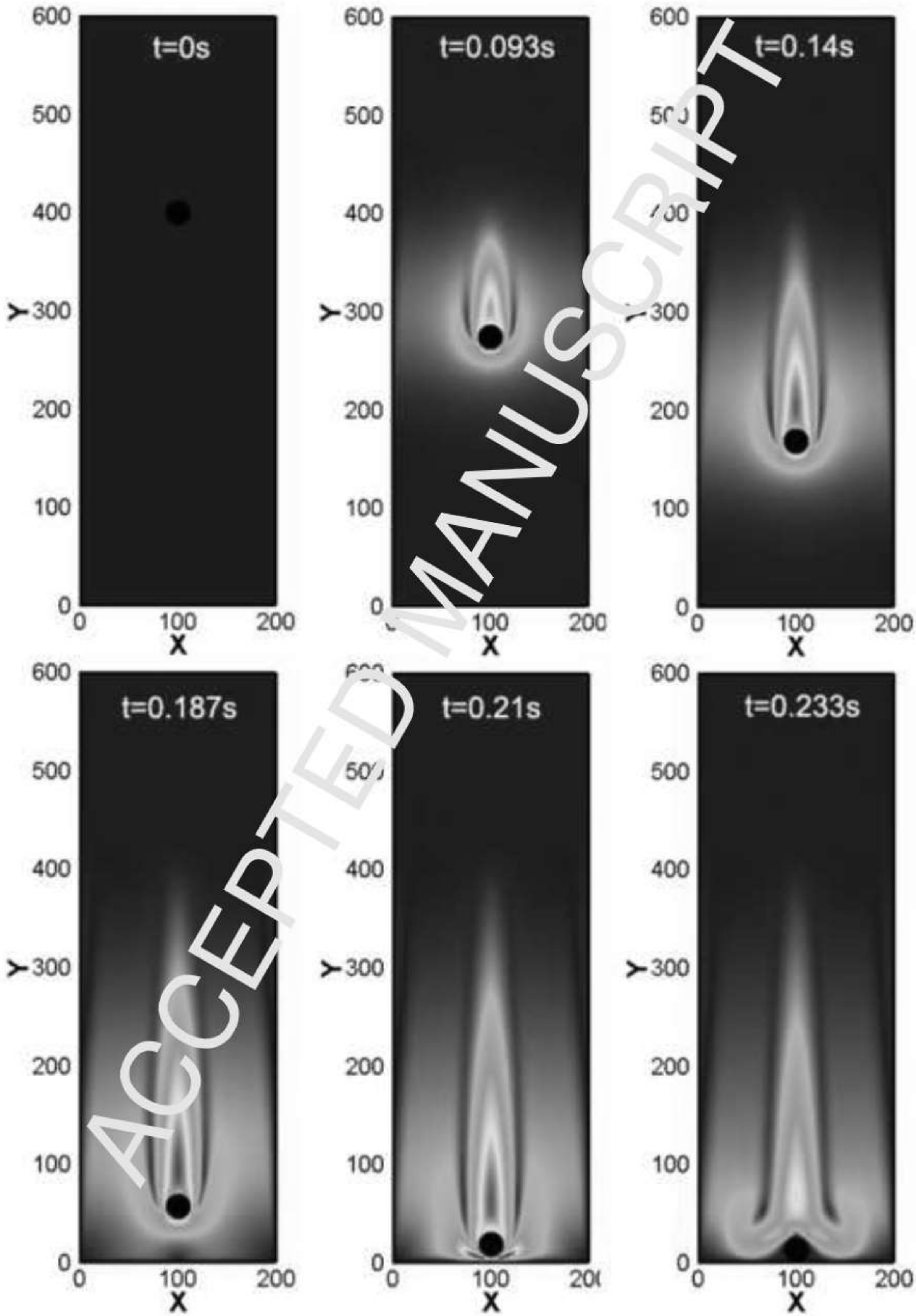


Fig. 14 Comparison of particle movement in Y direction

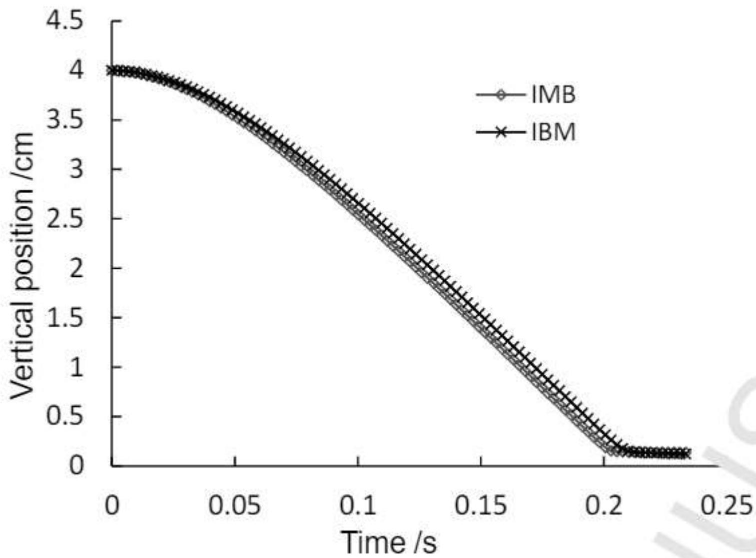


Fig. 15 Comparison of particle velocity in Y direction

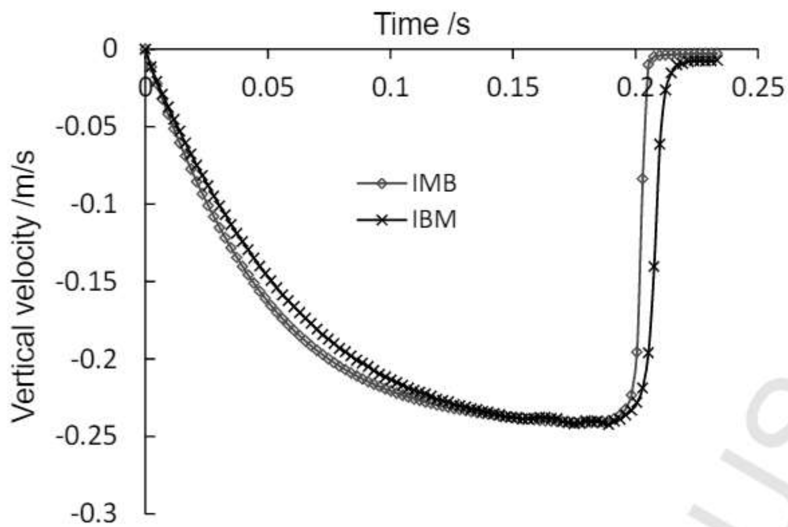


Fig. 16 Comparison of drag forces applied to particle

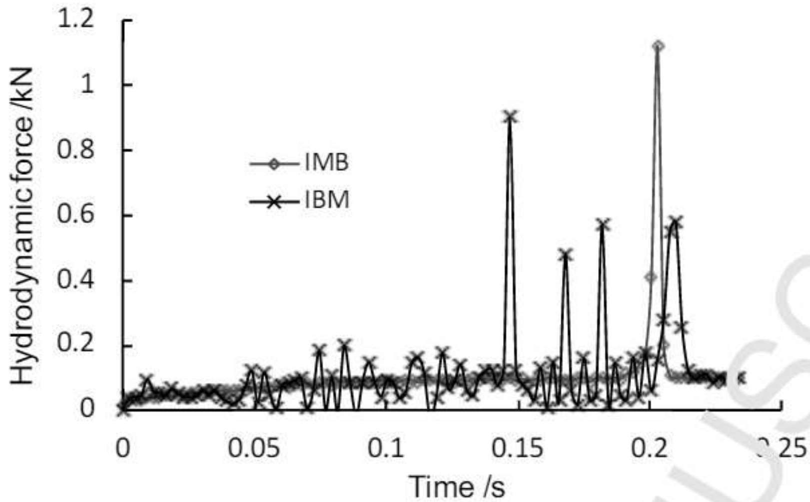


Fig. 17 Grid size effect on fluid-particle interaction

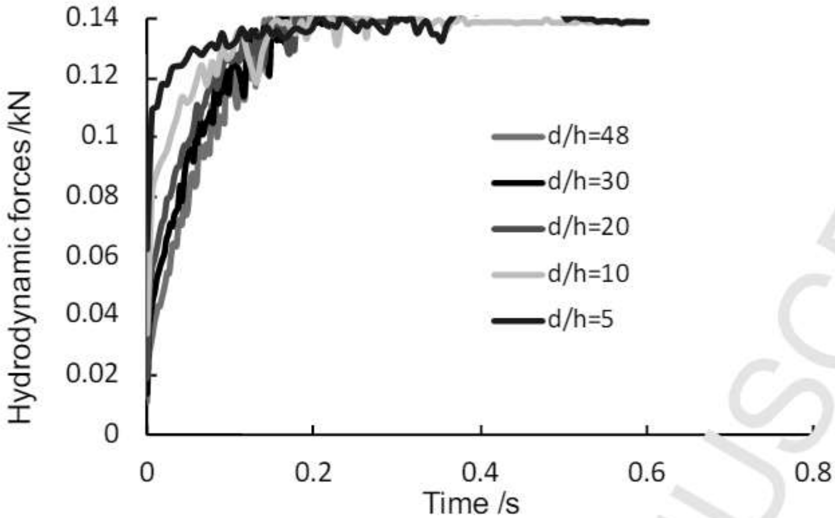


Table 1

Table 1 The increment value of movement

IZONE	IV	III	II	I
MARCH(X, IZONE)	-1	0	1	0
MARCH(Y, IZONE)	0	1	0	-1