

Numerical Simulations of Irregular Particle Transport in Turbulent Flows Using Coupled LBM-DEM

K. Han, Y. T. Feng and D. R. J. Owen¹

Abstract: Numerical procedures are introduced for simulations of irregular particle transport in turbulent flows using the coupled lattice Boltzmann method (LBM) and the discrete element method (DEM). The fluid field is solved by the extended LBM with the incorporation of the Smagorinsky turbulence approach, while particle interaction is modeled by the DEM. The hydrodynamic interactions between fluid and particles are realised through an immersed boundary condition, which gives rise to a coupled solution strategy to model the fluid-particle system under consideration. Main computational aspects comprise the lattice Boltzmann formulation for the solution of fluid flows; the incorporation of the large eddy simulation (LES) based turbulence model in the framework of the LBM for turbulent flows; the immersed boundary condition for hydrodynamic interactions between fluid and moving particles; and the DEM modelling of the interactions between irregular particles. As a demonstration of the applicability of the proposed methodology, a number of test cases are provided for polygonal and superquadric particle transport in fluid flows at high Reynolds numbers.

Keyword: Lattice Boltzmann method, Discrete element method, Fluid-particle interactions, Immersed boundary condition, Smagorinsky turbulence model

1 Introduction

The transport of solid particles within a fluid flow has a wide range of engineering applications, such as minerals recovery and food processing. The understanding of the underlying particle-fluid dy-

namics in the problem has been limited by the lack of powerful analysis tools. In a particle-fluid system, the fluid dynamics of the flow is influenced by the presence of solid particles, and the motion of the solid particles is, in turn, driven by fluid-induced forces.

In recent years, the lattice Boltzmann method (LBM) has emerged as an alternative to the conventional computational fluid dynamics (CFD) methods employing Navier-Stokes equations. It offers various advantages over the Navier-Stokes equations, including high space-time resolution, full scalability on parallel computers, as well as efficient and robust implementation in complex geometries Chen et. al (2003). Another distinct feature of the LBM over the finite volume method and finite element method is the use of an Eulerian grid to represent the flow field. For these reasons, the LBM is ideal for simulating fluid flows in complex geometries, such as particle transport. Since Ladd's early work Ladd (1994), the LBM has been widely employed to model fluid-particle interactions, see for instance Aidun et.al (1998) Qi and Luo (2003) Nguyen and Ladd (2005) Feng and Michaelides (2004) Feng and Michaelides (2005). Further employing the discrete element method (DEM) to account for particle-particle interactions gives rise to a combined LBM-DEM solution procedure. The explicit time stepping scheme of both LBM and DEM makes this coupling strategy a competitive numerical tool for the simulation of particle-fluid systems, having potential to be a powerful predictive tool for gaining fundamental insights into many poorly understood physical phenomena in the problem under consideration. Such a coupled methodology was first proposed by Cook et.al (2004) in simulating particle-fluid systems dominated by particle-fluid and particle-particle

¹ Civil and Computational Engineering Centre, School of Engineering, University of Wales Swansea, SA2 8PP, UK.

interactions.

While the LBM has been well established for laminar flows, turbulence modeling within the framework of the lattice Boltzmann equation (LBE) remains a challenge. Recently, limited attempts have been made to incorporate some existing turbulence models into the LBM. The conventional large eddy simulation (LES) with a one-parameter Smagorinsky sub-grid model Smagorinsky (1963) is the simplest to apply. The approach assumes that the Reynolds stress tensor is dependent only on the local strain rate. This model is extremely convenient in numerical simulation as it leaves the Navier-Stokes equation invariant except for a renormalised turbulent viscosity Latt et. al (2005).

In our previous work Han et. al (2007), the coupled LBM-DEM strategy has been successfully implemented in the simulation of circular particle transport problems. Nevertheless, non-circular particles are often used in practice to represent a more realistic situation. In this paper, we present numerical procedures for simulating polygonal and superquadric particle transport in turbulent flows.

The remainder of the paper is organised as follows. The next section gives a brief introduction to the LBM and its incorporation with the LES based Smagorinsky model. The modeling of fluid-particle and particle-particle interactions are discussed in Section 3. Finally a number of examples are provided in Section 4 for polygonal and superquadric particle transport in turbulent flows.

2 The Lattice Boltzmann Method

The lattice Boltzmann method (LBM) describes the fluid in terms of fluid particle density functions at discrete lattice and discrete time. It simulates fluid flows by tracking the evolution of fluid particle distributions instead of tracking single fluid particles. Once the distribution function is solved, the macroscopic variables, such as velocity and pressure, of the fluid field can be conveniently calculated from its first two moments.

In the LBM, space is divided into regular lattice nodes. The fluid is modeled as a group of fluid

particles that are allowed to move between lattice nodes or stay at rest. During each discrete time step of the simulation, fluid particles move to the nearest lattice node along their directions of motion, where they 'collide' with other fluid particles that arrive at the same node. The outcome of the collision is determined by solving the kinetic (Boltzmann) equation for the new distribution function at that node and the fluid particle distribution function is updated Chen and Doolen (1998).

2.1 D2Q9 model

The LBE with single-relaxation-time approximation introduced by Bhatnagar, Gross and Krook (BGK) for the collision operator is expressed as

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau} [f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)] \quad (1)$$

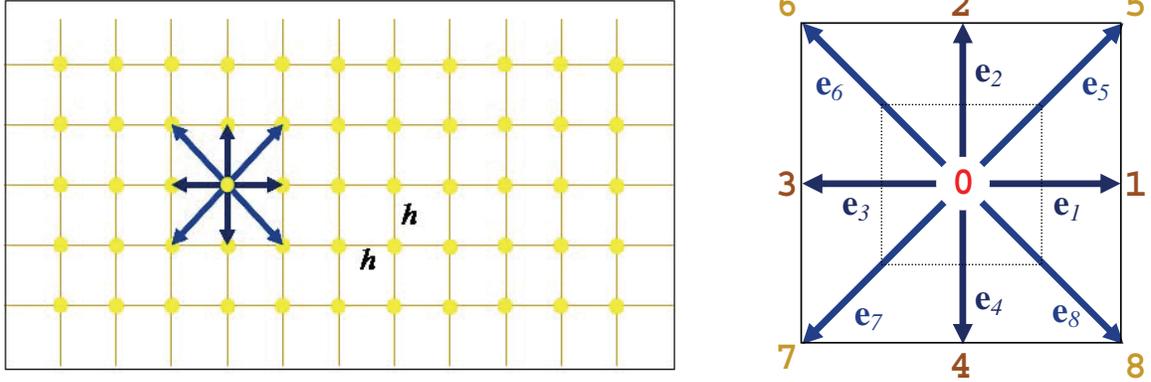
where f_i is the density distribution function with discrete velocity \mathbf{e}_i along the i -th direction; f_i^{eq} is the equilibrium distribution function; and τ is the single relaxation time which controls the rate of approach to equilibrium. The left-hand side of Eq.(1) denotes a streaming process for fluid particles while the right-hand side models the collisions through relaxation.

In the widely used D2Q9 model Qian et. al (1992), the fluid particles at each node move to their eight immediate neighbouring nodes with discrete velocities \mathbf{e}_i , ($i = 1, \dots, 8$), as shown in Fig. 1. A proportion of the particles can rest at the node, which is equivalent to moving with a zero velocity \mathbf{e}_0 . Referring to the numbering system shown in Fig. 1b, the nine discrete velocity vectors are given by

$$\begin{cases} \mathbf{e}_0 = (0, 0) \\ \mathbf{e}_1 = C(1, 0); & \mathbf{e}_2 = C(0, 1); \\ \mathbf{e}_3 = C(-1, 0); & \mathbf{e}_4 = C(0, -1) \\ \mathbf{e}_5 = C(1, 1); & \mathbf{e}_6 = C(-1, 1); \\ \mathbf{e}_7 = C(-1, -1); & \mathbf{e}_8 = C(1, -1) \end{cases} \quad (2)$$

where C is termed the lattice speed and defined as

$$C = h/\Delta t$$



(a) Lattice of the LBM

(b) D2Q9 model

Figure 1: Space discretisation and D2Q9 model

with h the lattice spacing and Δt the discrete time step.

The equilibrium distribution functions f_i^{eq} depend only on local density and velocity and are defined in D2Q9 model as

$$\begin{cases} f_0^{eq} = \rho \left(1 - \frac{3}{2C^2} \mathbf{v} \cdot \mathbf{v}\right) \\ f_i^{eq} = w_i \rho \left(1 + \frac{3}{C^2} \mathbf{e}_i \cdot \mathbf{v} + \frac{9}{2C^2} (\mathbf{e}_i \cdot \mathbf{v})^2 - \frac{3}{2C^2} \mathbf{v} \cdot \mathbf{v}\right) \end{cases} \quad (i = 1, \dots, 8) \quad (3)$$

in which w_i is the weighting factor defined as:

$$w_0 = \frac{4}{9}; \quad w_{1,2,3,4} = \frac{1}{9}; \quad w_{5,6,7,8} = \frac{1}{36} \quad (4)$$

The computation at each time step comprises two operations: *collision* and *streaming*. The first operation simulates fluid particle collisions, which cause the fluid particles at each lattice node to scatter into different directions. The collision rules are chosen to leave the sum of the density distribution functions unchanged, or no particle is lost. The rules are also chosen to conserve the total energy and momentum at each lattice node Chen and Doolen (1998). This computation is completely local. The second operation, streaming, is to advance the particles to the next lattice node along their directions of motion. The streaming operation takes little computational effort. These features make the LBM highly efficient, simple to implement and natural to parallelise.

The macroscopic fluid variables, density ρ and velocity \mathbf{v} , can be recovered from the distribution functions as

$$\rho = \sum_{i=0}^8 f_i \quad \rho \mathbf{v} = \sum_{i=1}^8 f_i \mathbf{e}_i \quad (5)$$

while the fluid pressure field p is determined by the following equation of state

$$p = C_s^2 \rho \quad (6)$$

where C_s is termed the fluid speed of sound and is related to the lattice speed C by

$$C_s = C/\sqrt{3} \quad (7)$$

The kinematic viscosity, ν , of the fluid is implicitly determined by the model parameters, h , Δt and τ as

$$\nu = \frac{1}{3} \left(\tau - \frac{1}{2} \right) \frac{h^2}{\Delta t} = \frac{1}{3} \left(\tau - \frac{1}{2} \right) Ch \quad (8)$$

which indicates that the selection of these three parameters has to be related to each other to achieve a correct fluid viscosity.

It can be proved that the LBE (1) recovers the incompressible Navier-Stokes equations to the second order in both space and time Chen and Doolen (1998), which is the theoretical foundation for the success of the LBM for modeling general fluid flow problems. However, since it is obtained by the linearised expansion of the original

kinetic theory based LBE, Eq. (1) is only valid for small velocities, or small 'computational' Mach number, M_a , defined by

$$M_a = \frac{v_{\max}}{C} \quad (9)$$

where v_{\max} is the maximum simulated velocity in the flow.

Generally smaller Mach number implies more accurate solution. It is therefore required that

$$M_a \ll 1 \quad (10)$$

i.e., the lattice speed C should be sufficiently larger than the maximum fluid velocity to ensure a reasonably accurate solution.

The positivity of the kinematic viscosity (8) requires that

$$\tau > 1/2$$

τ is also largely responsible for the numerical stability of the lattice Boltzmann modeling. Generally speaking, a larger value represents a more viscous fluid and the simulation is more stable, whilst a smaller value corresponds a less viscous fluid and the scheme is more prone to numerical instability, particularly when τ is approaching 0.5. This can also be understood from the fact that the limitation to the value of τ is imposed by the explicit feature of the lattice Boltzmann formulation. It may be of both theoretical and practical importance if the critical value of τ , up to which a stable solution can be achieved, is known. Theoretically, this value satisfies an equation but it is highly nonlinear and strongly dependent on the actual flow pattern. As a result, it is impractical to attain unless for a very simple flow case Sterling and Chen (1996).

2.2 Incorporating turbulence model in the LBM

While the LBM has been proven to be an efficient simulation tool for a variety of complex flow problems, the modeling of turbulent flows within the framework of the LBM is not a well investigated topic and only very limited work has been reported. However, many engineering applications are often associated with high Reynolds

numbers which are turbulent in nature. Therefore the incorporation of a turbulence model into the LBM is essential for simulating realistic particle transport problems under consideration.

The popular large eddy simulation (LES) aims at directly solving large spatial scale turbulent eddies which carry the major portion of the flow's energy, while modeling the smaller scale eddies using a sub-grid model. The separation of these scales is achieved through the filtering of the Navier-Stokes equations, from which the solutions to the resolved scales are directly obtained. Unresolved scales can be modeled by, for instance, the widely used one-parameter Smagorinsky sub-grid model Smagorinsky (1963) that assumes that the Reynolds stress tensor is dependent only on the local strain rate.

To incorporate the LES in the LBM, Eq.(1) has to be modified to include the eddy viscosity, which is realised by the approach described in Yu et. al (2005).

The filtered form of the LBE Yu et. al (2005) is expressed as

$$\begin{aligned} \tilde{f}_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) \\ = \tilde{f}_i(\mathbf{x}, t) - \frac{1}{\tau_*} [\tilde{f}_i(\mathbf{x}, t) - \tilde{f}_i^{eq}(\mathbf{x}, t)] \quad (11) \end{aligned}$$

where \tilde{f}_i and \tilde{f}_i^{eq} represent respectively the distribution function and the equilibrium distribution function at the resolved scale. The effect of the unresolved scale motion is modeled through an effective collision relaxation time scale τ_r . Thus in Eq.(11) the total relaxation time should be

$$\tau_* = \tau + \tau_r$$

where τ and τ_r are respectively the relaxation times corresponding to the true fluid (molecular) viscosity ν and the turbulence viscosity ν_* defined by a sub-grid turbulence model. Accordingly ν_* is given by

$$\begin{aligned} \nu_* = \nu + \nu_t &= \frac{1}{3}(\tau_* - \frac{1}{2})C^2\Delta t \\ &= \frac{1}{3}(\tau + \tau_r - \frac{1}{2})C^2\Delta t = \frac{1}{3}\tau_r C^2\Delta t \end{aligned}$$

By employing the Smagorinsky model, the turbulence viscosity ν_t is explicitly calculated from the

filtered strain rate tensor $\tilde{S}_{ij} = (\partial_j \tilde{u}_i + \partial_i \tilde{u}_j)/2$ and a filter length scale (which is equal to the lattice spacing h) as

$$v_t = (S_c h)^2 \hat{S} \quad (12)$$

where S_c is the Smagorinsky constant; and \hat{S} the characteristic value of the filtered strain rate tensor \tilde{S}

$$\hat{S} = \sqrt{\sum_{i,j} \tilde{S}_{ij} \tilde{S}_{ij}}$$

An attractive feature of the model is that \tilde{S} can be obtained directly from the second-order moments, \tilde{Q} , of the non-equilibrium distribution function

$$\tilde{S} = \frac{\tilde{Q}}{2\rho S_c \tau_*} \quad (13)$$

in which \tilde{Q} can be simply computed by the filtered density functions at the lattice nodes

$$\tilde{Q}_{ij} = \sum_{k=1}^8 e_{ki} e_{kj} (\tilde{f}_k - \tilde{f}_k^{eq}) \quad (14)$$

where e_{ki} is the k -th component of the lattice velocity \mathbf{e}_i . Consequently

$$\hat{S} = \frac{\hat{Q}}{2\rho S_c \tau_*} \quad (15)$$

with \hat{Q} the filtered mean momentum flux computed from \tilde{Q}

$$\hat{Q} = \sqrt{2 \sum_{i,j} \tilde{Q}_{ij} \tilde{Q}_{ij}} \quad (16)$$

The above approach is extremely convenient in terms of numerical implementations as it leaves the LBE unchanged except for the use of a turbulent-related viscosity τ_* . With this extended LBM for turbulent flows, good results have been reported for simulations of a well documented benchmark test at $Re = 40000$ Rodi et. al (1997); Krafczyk and Toelke (2003). Yu Yu et. al (2005) *et al* indicates that this model can accurately capture important features of the decaying homogeneous isotropic turbulence and is potentially a reliable computational tool for turbulence simulations.

3 Modeling of Particle-Fluid and Particle-Particle Interactions

3.1 Hydrodynamic forces for fluid-particle interactions

For particle transport problems concerned, the modeling of the interaction between fluid and solid particle requires a physically correct 'no-slip' velocity condition imposed on their interface. In other words, the fluid adjacent to the particle surface should have identical velocity as that of the particle surface. For a stationary solid particle, this 'no-slip' velocity condition can be easily achieved at the fluid-particle interface by the bounce-back scheme. Assume that a solid particle is mapped onto the lattice by a set of lattice nodes. The nodes inside and outside the solid region are respectively termed solid nodes and fluid nodes. If i is a link (or direction) between a fluid node and a solid node, the bounce-back rule states that the incoming fluid particle from the fluid node is reflected back to the node it comes from, i.e.

$$f_{-i}(\mathbf{x}, t+1) = f_i(\mathbf{x}, t_+) \quad (17)$$

where $f_i(\mathbf{x}, t_+)$ denotes the post collision distribution at the boundary node \mathbf{x} , and $-i$ is the opposite direction of i . This simple rule ensures that no tangential velocity exists along the fluid-solid interface, therefore a 'no-slip' condition is imposed. Note that the solid particle boundary is assumed to be situated halfway between the fluid and solid node so as to achieve a second order accuracy. Otherwise, the accuracy is of first order.

It is, however, not trivial to model the interaction between the fluid and a moving particle. Ladd Ladd (1994) proposes a modification to the original bounce-back rule so that the movement of a solid particle can be accommodated. This approach provides a relationship of the exchange of momentum between the fluid and the solid boundary nodes. It also assumes that the fluid fills the entire volume of the solid particle, or in other words, the particle is modeled as a 'shell' filled with fluid. As a result, both solid and fluid nodes on either side of the boundary surface are treated in an identical fashion.

For a given boundary link i , the modified 'no-slip'

rule is given by

$$f_{-i}(\mathbf{x}, t+1) = f_i(\mathbf{x}, t_+) - \alpha_i \mathbf{e}_i \cdot \mathbf{v}_b \quad (\alpha_i = 6w_i \rho / C_s^2) \quad (18)$$

where \mathbf{v}_b is the velocity of the middle of the boundary link i and computed by

$$\mathbf{v}_b = \mathbf{v}_c + \boldsymbol{\omega} \times (\mathbf{x} + \mathbf{e}_i \Delta t / 2 - \mathbf{x}_c)$$

in which \mathbf{v}_c and $\boldsymbol{\omega}$ are respectively the translational and angular velocities of the solid particle; \mathbf{x}_c and $\mathbf{x} + \mathbf{e}_i \Delta t / 2$ are respectively the particle centre and mid-boundary link coordinates. The modified rule is illustrated in Fig. 2.

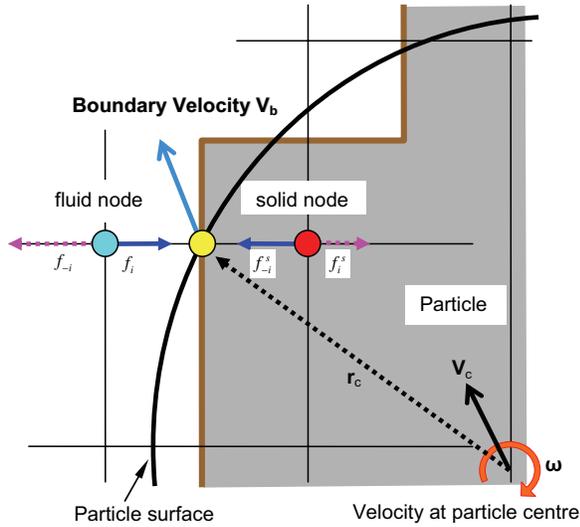


Figure 2: Modified bounce-back rule for moving particle

The hydrodynamic force and torque exerted on the solid particle at the boundary node are computed as

$$\mathbf{F}_i = 2 [f_i(\mathbf{x}, t_+) - \alpha_i \mathbf{e}_i \cdot \mathbf{v}_b] / \Delta t \quad (19)$$

$$\mathbf{T}_i = \mathbf{r}_c \times \mathbf{F}_i \quad (\mathbf{r}_c = \mathbf{x} + \mathbf{e}_i \Delta t / 2 - \mathbf{x}_c) \quad (20)$$

Then the total hydrodynamic forces and torque exerted on the solid particle are computed by summing up the forces and torques from all the related boundary links as

$$\mathbf{F} = \sum_i \mathbf{F}_i; \quad \mathbf{T} = \sum_i \mathbf{T}_i \quad (21)$$

It has been observed, however, that the computed hydrodynamic forces may suffer from severe fluctuations when the particle moves across the grid with a large velocity. This is mainly caused by the stepwise representation of the solid particle boundary and the constant changing boundary configurations.

To circumvent the fluctuation of the computed hydrodynamic forces with the modified bounce-back rule, Noble and Torczynski (1998) put forward to an immersed moving boundary (IMB) method. In this approach, a control volume is introduced for each lattice node that is a $h \times h$ square around the node, as illustrated by the shadow area in Fig. 3a. Meanwhile, a local fluid to solid ratio γ is defined, which is the volume fraction of the nodal cell covered by the solid particle as shown in Fig. 3b.

The LBE for those lattice nodes (fully or partially) covered by a solid particle is modified to enforce the 'no-slip' velocity condition as

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau} (1 - \beta) [f_i(\mathbf{x}, t) - f_i^{eq}] + \beta f_i^m \quad (22)$$

where β is a weighting function depending on the local fluid/solid ratio γ ; and f_i^m is an additional term that accounts for the bounce back of the non-equilibrium part of the distribution function, computed by the following expressions

$$\begin{cases} \beta = \frac{\gamma(\tau-0.5)}{(1-\gamma)+(\tau-0.5)} \\ f_i^m = f_{-i}(\mathbf{x}, t) - f_i(\mathbf{x}, t) + f_i^{eq}(\rho, \mathbf{v}_b) - f_{-i}^{eq}(\rho, \mathbf{v}) \end{cases} \quad (23)$$

The total hydrodynamic forces and torque exerted on a solid particle over n particle-covered nodes are summed up as

$$\mathbf{F}_f = Ch \left[\sum_n \left(\beta_n \sum_i f_i^m \mathbf{e}_i \right) \right] \quad (24)$$

$$\mathbf{T}_f = Ch \left[\sum_n (\mathbf{x}_n - \mathbf{x}_c) \times \left(\beta_n \sum_i f_i^m \mathbf{e}_i \right) \right] \quad (25)$$

where \mathbf{x}_n is the coordinate of the lattice node n .

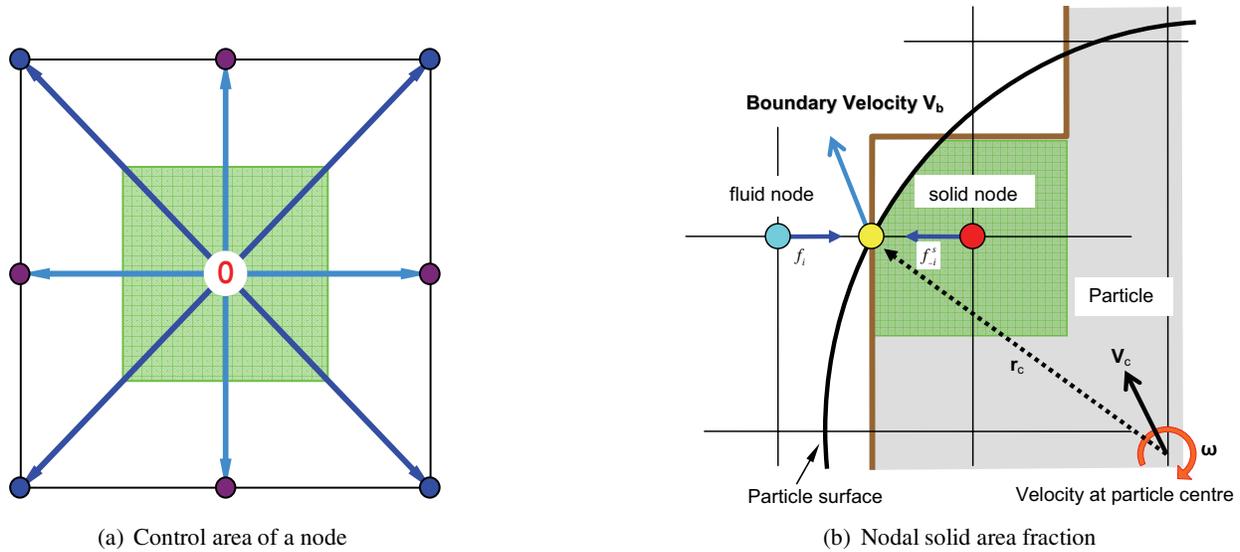


Figure 3: Immersed boundary scheme of Noble and Torczynski Noble and Torczynski (1998)

With this approach, the computed hydrodynamic forces are sufficiently smooth, which is also confirmed by our numerical tests.

3.2 Contact forces for particle-particle interactions

In many lattice Boltzmann simulations of particle-fluid problems, the inter-particle interaction is either ignored or simply treated. However, many practical applications requires accurate resolutions of the particle contact. A rational choice is to employ the discrete element method (DEM) to account for this interaction.

The DEM, originated in geotechnical and granular flow applications in the 70's by Cundall and Strack Cundall and Strack (1979), has now become a promising numerical tool capable of simulating problems of a discrete or discontinuous nature. In its classical form, a discrete system is considered as an assembly of individual discrete objects which are treated as rigid and represented by discrete elements as simple geometric entities. The dynamic response of discrete elements is largely determined by the interaction laws specified for interelement contact and obeys Newton's second law. The contact interaction laws are normally developed within the framework of penalty methods which only approximately satisfies the contact displacement constraints, thereby allow-

ing a small amount of overlap to occur. The dynamic equations governing the evolution of the system are solved by the explicit central difference integration algorithm that is consistent with the lattice Boltzmann formulation.

Disks and spheres are the most commonly used discrete objects in the discrete element simulation, mainly due to their geometric simplicity and computational efficiency. However, they completely lack angularity, or in other words, they cannot provide resistance to rolling motion. It has been increasingly recognised that the angularity of discrete objects plays an important role in simulating dynamic behaviours of many practical problems. Consequently some non-circular discrete objects are introduced, such as polygons (non-smooth boundary) and superquadrics (smooth boundary).

Polygon is one of the basic geometric entities, but its application in the DEM has, however, been limited. The underlying difficulty is the proper handling of corner-corner contact in which gap functions and normal directions are not well defined. In the commonly used node-segment contact models the direction of normal force may exhibit a discontinuity when the contact pair is evolving. Recently a simple non-ambiguous corner-corner contact algorithm has been developed by Feng and Owen Feng and Owen (2004)

within an energy based theoretical framework, in which the normal and tangential directions, magnitude and reference contact position of the contact forces are uniquely defined.

Conventionally a superquadric may be preferred over a polygon since it can represent a wide range of shapes with less information. In addition, superquadric-superquadric contact does not exist corner-corner contact situation, as is the case in polygon-polygon contact. Nevertheless, superquadric, though mathematically elegant, is not widely used as substantial computational cost is involved. For instance, an overlap check between two superquadrics requires the solution of the intersections of two nonlinear functions, which is a very expensive operation and may sometimes fail in finding the solution.

With the corner-corner contact algorithm Feng and Owen (2004), the contact between two polygons is thus resolved, which also promotes the development of a robust and efficient polygon-based contact resolution for superquadrics. Firstly, a superquadric is approximated with a convex polygon through adaptive sampling. Secondly, by clipping two polygons, an efficient linear algorithm is performed to search for intersections and overlap area of the polygons. Finally, the contact forces and directions are uniquely determined. The procedure is outlined below. More details can be found in Han et. al (2006).

3.2.1 Polygonal approximation of a superquadric

A 2D superquadric is defined by the following implicit function

$$f(x, y) = \left(\frac{x}{a}\right)^m + \left(\frac{y}{b}\right)^m - 1 = 0 \quad (26)$$

where parameters a , b , and m (integer) determine the shape of the superquadric. If $m = 2$, an ellipse or disk (with $a = b$) is recovered. By varying a and b , the represented object can be stretched in the x and y directions. A superquadric becomes more 'blocky' as m increases from 2 to infinity.

When a superquadric is used as a discrete element, a common practice is to approximate it with a convex polygon, which is equivalent to approxi-

imating the superquadric curve with piecewise linear segments. An adaptive sampling procedure is employed in this work in order to achieve a better polygonal representation of the superquadric. Generally a superquadric may be approximated by either an 'inscribed' or 'circumscribed' polygon, but the former is more rational and simpler to generate. The sampling procedure is omitted here.

Once all the superquadrics are represented with polygons, a contact detection algorithm is adopted to identify the pairs with potential contact. Subsequently the contact resolution is performed, by searching for intersections of the contacting polygons first, followed by employing the polygon-polygon contact model.

3.2.2 Intersections of two polygons

Assume that two polygons, P and Q , to be considered have n and m edges/vertices respectively. A naive algorithm for resolving the intersections of the two polygons can be readily formulated by sequentially checking the intersection of each edge of one polygon with each edge of the other polygon, and its order of complexity is thus nm .

A linear algorithm is proposed by O'Rourke *et al* (1982), where two special pointers, distinguishing one edge on each polygon, are maintained. The pointers are advanced around the polygons such that their edges 'chase' one another, searching for the intersection points. All the intersection points can be found within two cycles around the polygons, and thus the algorithm achieves a linear complexity of $2(n + m)$.

This work adopts the linear algorithm in a modified form. In fact, the overlap of two polygons only occurs in a very small zone in the discrete element simulation. Hence if the possible overlap zone can be identified, the searching for intersections may be performed only for the edges lying in the zone. This is achieved through the following two steps:

1. The possible overlap zone (rectangle) is determined by the overlap area of the bounding boxes of two polygons under consideration;

2. By clipping the polygons against this zone and discarding edges lying outside of the zone, the linear algorithm described in Rourke et. al (1982) is employed to search for intersections of the two polygons.

Polygon clipping against a rectangular window is a fundamental operation in computer graphics and a number of effective algorithms have been proposed. The Sutherland-Hodgman polygon-clipping algorithm is employed and slightly modified in the sense that only those edges with two ends lying outside of the clipping zone are excluded. Due to the small clipping (overlap) zone, the two polygons are likely to be clipped into two polylines with very few edges.

Note that with the above modification two polygons without overlap may be detected prior to the intersection solution procedure, if the bounding boxes of polygons do not overlap; or if one of the polygons lies completely outside of the clipping zone.

3.2.3 Contact forces and directions

Polygon-polygon contact is a typical corner-corner contact problem, or its special case, such as corner-edge contact, or its extensions for more complex contact situations. The corresponding contact forces and directions are computed by utilising the corner-corner contact model developed by Feng and Owen (2004). It is assumed that the contact of two polygons is associated with a contact energy function. The contact line that connects the two intersections of the polygons plays a crucial role. The direction of the line defines the tangential direction and the normal direction is perpendicular to it. The middle point of the line is identified as the reference contact point where the normal force should be applied. The commonly used overlap distance/gap is not explicitly present in the model. Instead, its usual role is replaced by the length of the contact width, which is a well defined characteristic contact length. The model in its final form is simple and elegant with a clear geometric perspective, and also possesses some advanced features. The algorithmic aspects can be found in

Feng and Owen (2004).

3.3 LBM and DEM Coupling

Fluid and solid particle coupling at each time step is realised by first computing the fluid solution, and then updating the particle positions through the integration of the equations of motion given by

$$\begin{cases} m\mathbf{a} + c\mathbf{v} = \mathbf{F}_c + \mathbf{F}_f + m\mathbf{g} \\ J\ddot{\theta} = \mathbf{T}_c + \mathbf{T}_f \end{cases} \quad (27)$$

where m and J are respectively the mass and the moment of inertia of the solid particle; $\ddot{\theta}$ the angular acceleration; \mathbf{g} the gravitational acceleration if considered; \mathbf{F}_f and \mathbf{T}_f are respectively the hydrodynamic force and torque; \mathbf{F}_c and \mathbf{T}_c denote the contact force and torque from other particles and/or boundary walls; c is a damping coefficient; and the term $c\mathbf{v}$ represents a viscous force that accounts for the effect of all possible dissipation forces in the system. The static buoyancy force of the fluid is taken into account by reducing the gravitational acceleration to $(1 - \rho/\rho_s)\mathbf{g}$, where ρ_s is the density of a particle.

This dynamic equation governing the evolution of the system can be solved by the central difference scheme. Some important computational issues regarding the solution are briefly discussed as follows.

(1). Subcycling time integration. There are two time steps used in the coupled LBM-DEM procedure, Δt for the fluid flow and Δt_D for the particles. Since Δt_D is generally smaller than Δt , it has to be reduced to Δt_s so that the ratio between Δt and Δt_s is an integer n_s :

$$\Delta t_s = \frac{\Delta t}{n_s} \quad (n_s = \lceil \Delta t / \Delta t_D \rceil + 1) \quad (28)$$

where $\lceil \cdot \rceil$ denotes an integer round-off operator. This basically gives rise to a so-called subcycling time integration for the DEM part; in one step of the fluid computation, n_s sub-steps of integration are performed for Eq. (27) using the time step Δt_s ; whilst the hydrodynamic forces \mathbf{F}_f and \mathbf{T}_f are kept unchanged during the subcycling.

(2). The dynamic equation in the lattice coordinate system. Since the LBE is implemented in

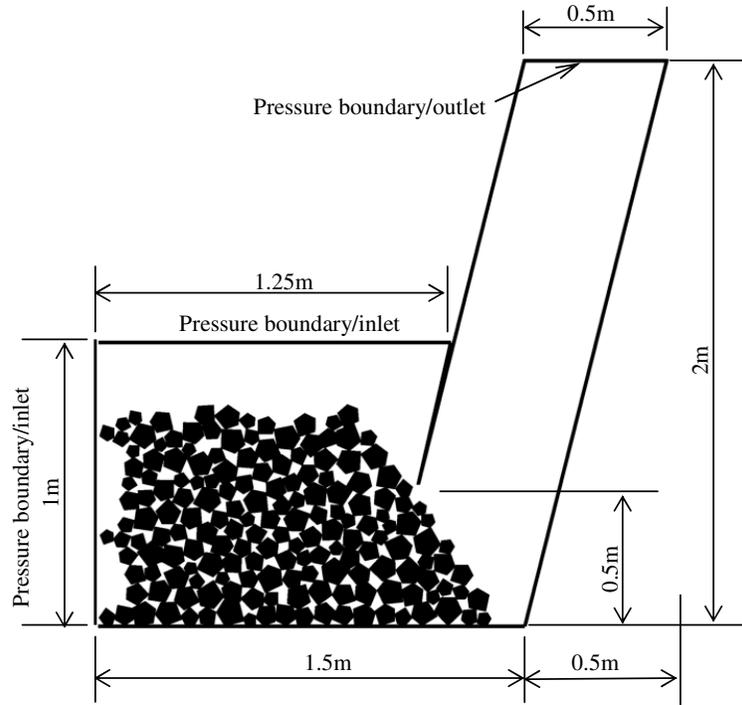


Figure 4: Problem description

the lattice coordinate system in this work, the dynamic equation Eq. (27) should be implemented in the same way. It can be derived that in the lattice coordinate system Eq. (27) takes the form of

$$\overline{m}\overline{\mathbf{a}} + \overline{c}_d\overline{\mathbf{v}} = \overline{\mathbf{F}}_c + \overline{\mathbf{F}}_f + \overline{m}\overline{\mathbf{g}} \quad (29)$$

where

$$\begin{cases} \overline{m} = m/\rho_s h^2 & \overline{\mathbf{v}} = \mathbf{v}/C \\ \overline{\mathbf{a}} = \mathbf{a}\Delta t/C; & \overline{\mathbf{g}} = \mathbf{g}\Delta t/C \\ \overline{c}_d = Chc_d; & \overline{\mathbf{F}}_t = \mathbf{F}_t/(\rho_0 C^2 h) \end{cases}$$

4 Numerical Illustrations

To illustrate the performance of the coupled LBM-DEM approach, test cases are provided for the simulations of polygonal and superquadric particle transport along a pipe under the suction action generated by the pressure difference at the inlet and outlet of the fluid domain.

The initial conditions are illustrated in Fig. 4, where the two inclined lines represent the pipe boundaries. The fluid domain is divided into a 800×800 square lattice with spacing $h = 2.5\text{mm}$. Though the fluid domain should be rectangular

in the LBM, a polygonal fluid domain is taken as the actual computational domain to reduce the computational costs, since both left-top and right-bottom sub-domains can be excluded from the simulation. To accommodate this irregularity, the actual domain profile is identified first, and the LBE (1) is applied only to the nodes within the profile. This is a generic approach which can be extended to any problem with an irregular exterior domain boundary. The material properties of the fluid are chosen as: density $\rho = 1000\text{kg/m}^3$ and kinematic viscosity $\nu = 5 \times 10^{-5}\text{m}^2/\text{s}$.

A constant pressure boundary condition with $\rho_{in} = \rho$ is imposed to the two (inlet) boundaries as shown in the figure. A smaller pressure with $\rho_{out} = 0.97\rho$ is applied to the outlet of the pipe. The remaining boundaries are assumed stationary walls and thus the 'no-slip' velocity condition is imposed.

The three examples comprise respectively 173 5-sided polygons, 158 ellipses (superquadrics of $m = 2$) and 70 superquadrics of $m = 5$, of different sizes positioned at the bottom of the domain, using the packing algorithms developed in Feng

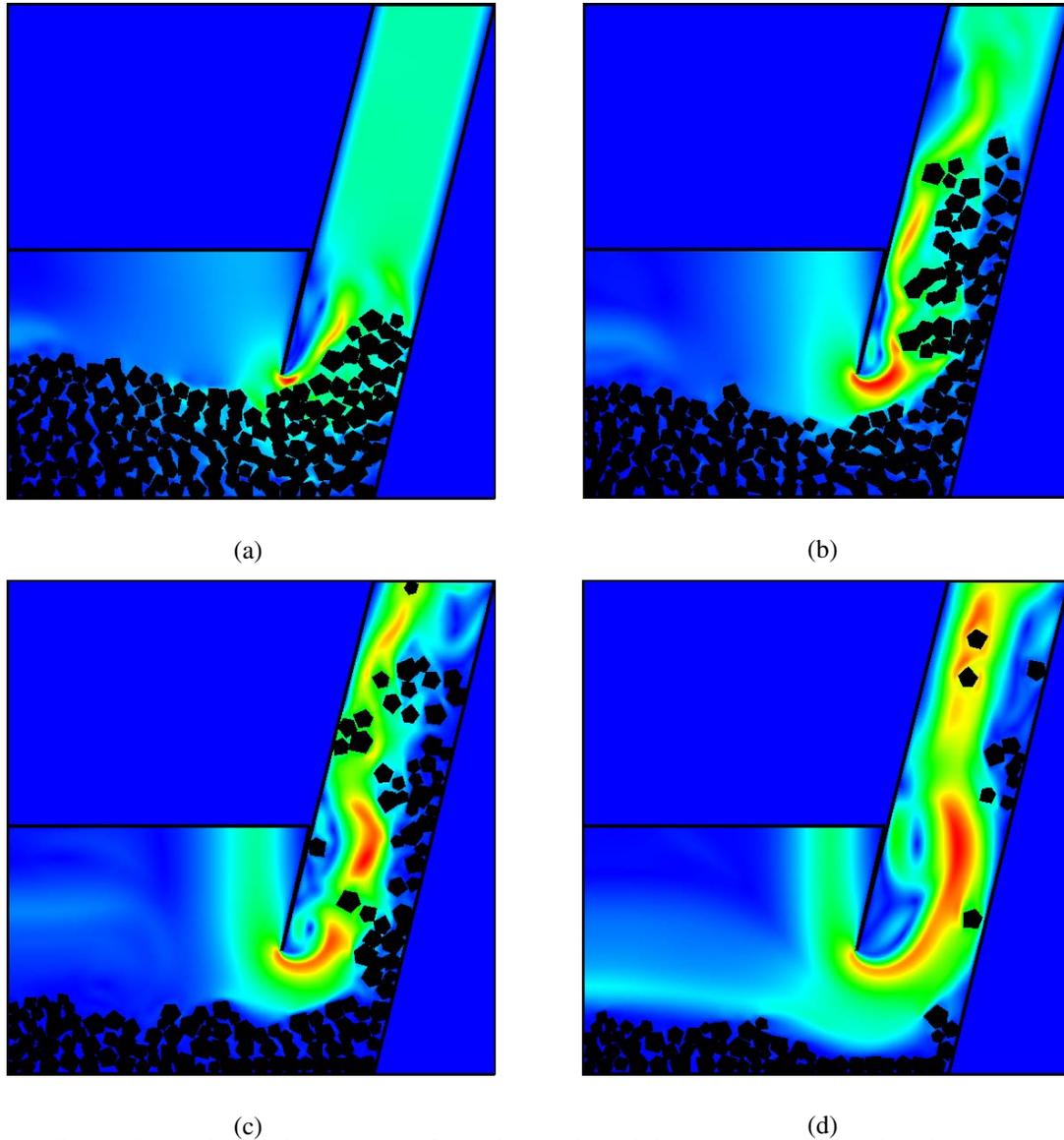


Figure 5: Total velocity contours for polygonal particle transport at four time instances

et.al (2003, 2002). Full gravity ($g = 9.81m/s^2$) is considered. The immersed boundary method of Noble and Torczynski Noble and Torczynski (1998) is employed to compute the fluid-particle interaction forces. The following parameters are chosen: particle density $\rho_s = 5000kg/m^3$, normal contact stiffness $k_n = 5 \times 10^8 N/m$, contact damping ratio $\xi = 0.5$ and time step factor $\lambda = 0.1$, which gives a time step of $\Delta t_d = 3.37 \times 10^{-5}$ for the DEM simulation of the particles.

The LES based one-parameter Smagorinsky turbulence model, as described in Section 3, is adopted with the Smagorinsky constant $S_c = 0.1$.

A complete simulation is achieved with $\tau = 0.501$. This gives a time step $\Delta t = 4.17 \times 10^{-5}s$ and thus the corresponding lattice speed $C = 60m/s$. The simulated maximum fluid velocities are respectively $v_{max} = 7.0m/s$ for polygons, $6.8m/s$ for ellipses and $5.9m/s$ for superquadrics at the pipe outlet (with the characteristic length $L = 0.5m$). The maximum Mach number and Reynolds number, for superquadric transport for example, are therefore estimated as

$$Ma = \frac{v_{max}}{C} = 0.0983; \quad Re = \frac{v_{max} * L}{\nu} = 59000$$

The Mach number indicates that the results ob-

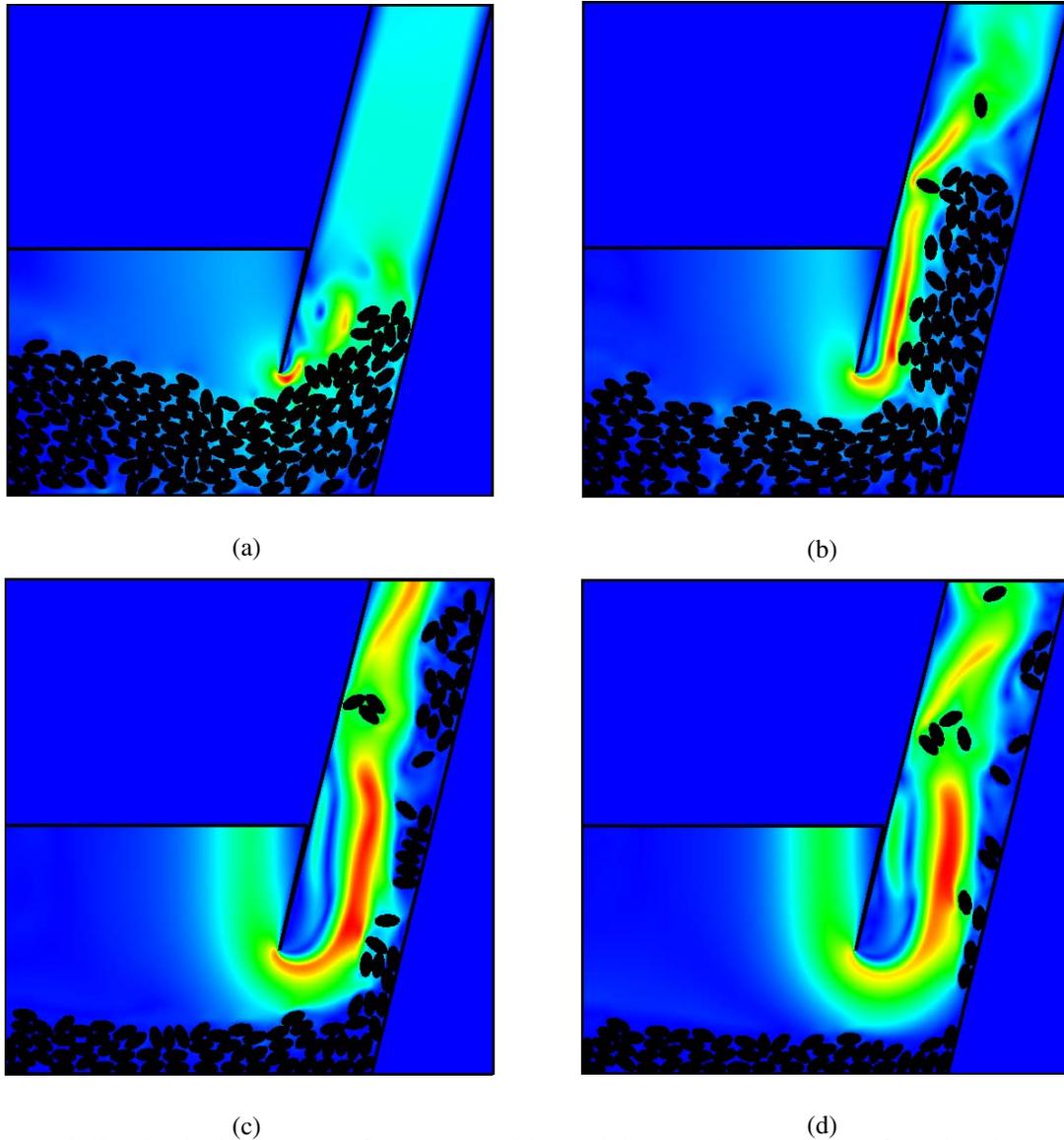


Figure 6: Total velocity contours for superquadratic particle ($m = 2$) transport at four time instances

tained are reasonably accurate. The flow field in terms of the total velocity contour and the evolution of the particles at four time instances are depicted in Figs.5-7(a)-(d), from which the complexity of the fluid field due to the particle-fluid and particle-particle interactions are clearly shown.

5 Conclusions

This paper introduces a coupled LBM-DEM solution strategy for numerical simulations of irregular particle transport in turbulent flows. The

test cases demonstrate that the proposed approach is a promising solution tool for the solution of particle-fluid interaction problems dominated by the presence of a large number of densely packed irregular particles and the occurrence of turbulence. The coupled approach appears to be robust (up to a certain Reynolds number) and the implementation is simple. Further tests indicate that the number of particles and their size distribution can be arbitrarily specified without causing any numerical problem. In addition, the proposed methodology can also be extended to 3D cases.

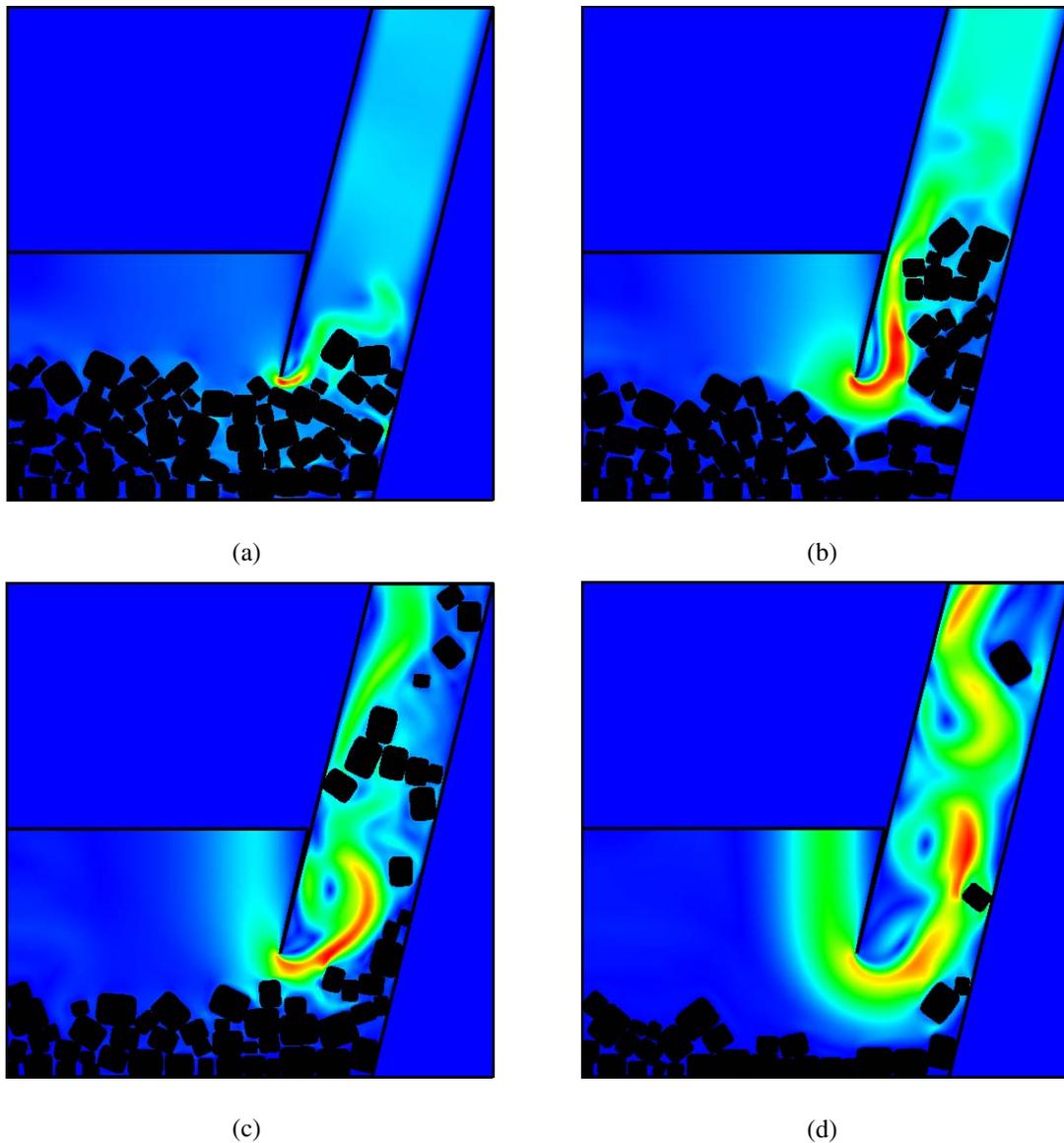


Figure 7: Total velocity contours for superquadric particle ($m = 5$) transport at four time instances

References

C. K. Aidun, Y. Lu, E.-G Ding. (1998): Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation. *Journal Fluid Mechanics*, 373, 287-311.

H. Chen, S. Kandasamy, S. Orszag, R. Shock, S. Succi, V. Yakhot. (2003): Extended Boltzmann kinetic equation for turbulent flows. *Science*, 301: 633-636.

S. Chen, G. Doolen. (1998): Lattice Boltzmann method for fluid flows. *Annual Review of Fluid*

Mechanics, 30, 329-364.

B. K. Cook, D. R. Noble, J. R. Williams. (2004): A direct simulation method for particle-fluid systems. *Engineering Computations*, 21(2-4):151-168.

P. A. Cundall, O. D. L. Strack. (1979): A discrete numerical model for granular assemblies. *Geotechnique*, 29(1):47-65.

Y. T. Feng, K. Han and D. R. J. Owen. (2003): Filling domains with disks: An advancing front

approach. *Int. J. Numer. Meth Engng.*, 56: 699-713.

Y. T. Feng, K. Han and D. R. J. Owen. (2002): An advancing front packing of polygons, ellipses and spheres. Proceedings of 3rd Int. Conf. on Discrete Element Methods, B. K. Cook & R. P.Jensen (ed.): pp93-98.

Y. T. Feng and D. R. J. Owen. (2004): A 2D polygon/polygon contact model: algorithmic aspects. *Engineering Computations*, 21: 265-277.

Z. G. Feng and E. E. Michaelides. (2004): The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. *Journal of Computational Physics*, 195: 602-628.

Z. G. Feng and E. E. Michaelides. (2005): Proteus: a direct forcing method in the simulations of particulate flows. *Journal of Computational Physics*, 202: 20-51.

K. Han, Y. T. Feng and D. R. J. Owen. (2006): Polygon-based contact resolution for superquadrics. *Int. J. Numer. Meth Engng.*, 66: 485-501.

K. Han, Y. T. Feng and D. R. J. Owen. (2007): Coupled lattice Boltzmann and discrete element modeling of fluid-particle interaction problems. *Computers and Structures*, (in press).

M. Krafczyk, J. Toelke. (2003): Large-eddy simulations with a multiple-relaxation-time LBE model. *I.J. Modern Physics B*, 17(1 & 2): 33-39.

Krafczyk, J. Toelke, J. Kuipers, K. van Duin, F. van Beckum, W. van Swaaij. (1992): A numerical model of gas-fluidized beds. *Chemical Engineering Science*, 47: 1913-1924.

A. Ladd. (1994): Numerical simulations of fluid particulate suspensions via a discretized Boltzmann equation (Parts I & II). *Journal Fluid Mechanics*, 271, 285-339.

A. Ladd, R. Verberg. (2001): Lattice-Boltzmann simulations of particle-fluid suspensions. *Journal of Statistical Physics*, 104 (5/6): 1191-1251.

J. Latt, B. Chopard, S. Succi, F. Toschi. (2005): Numerical analysis of the averaged flow field in a turbulent lattice Boltzmann simulation. *Physica A*.

N. Nguyen, A. Ladd. (2005): Sedimentation of hard-sphere suspensions at low Reynolds number. *Journal Fluid Mechanics*, 525, 73-104.

D. Noble, J. Torczynski. (1998): A lattice Boltzmann method for partially saturated cells. *International Journal of Modern Physics C*, 9, 1189-1201.

D. Qi, L. Luo. (2003): Rotational and orientational behaviour of three-dimensional spheroidal particles in Couette flows. *Journal Fluid Mechanics*, 477, 210-213.

Y. Qian, D. d'Humieres, P. Lallemand. (1992): Lattice BGK models for Navier-Stokes equation. *Europhys. Lett.* 17: 479-484.

W. Rodi, J. H. Ferziger, M. Breuer. (1997): Status of large eddy simulation: Results of a workshop. *ASME J. Fluids Eng.*, 119(2): 248-262.

J. O'Rourke, Chi-Bin Chien, T. Olson and D. Naddor. (1982): A new linear algorithm for intersecting convex polygons. *Computer Graphics and Image Processing*, 19:384-391.

J. Smagorinsky. (1963): General circulation model of the atmosphere. *Weather Rev.*, 99-164.

J. D. Sterling and S. Chen. (1996): Stability analysis of lattice Boltzmann methods. *Journal of Computational Physics*, 123:196-206.

T. E. Tezduyar. (2001): Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods In Engineering*, 8(2): 83-130.

H. Yu, S. Girimaji, L. Luo. (2005): DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *Journal of Computational Physics*, 209, 599-616.