



Lattice Boltzmann Simulation of Solid Particles Motion in a Three Dimensional Flow using Smoothed Profile Method

B. Khalili¹, M. Rahnama^{1†}, S. Jafari², F. Gharibi¹ and E. Jahanshahi Javaran³

¹ Mechanical Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran

² Petroleum Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran

³ Department of Energy, Institute of Science, High Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Iran.

†Corresponding Author Email: Rahnama@uk.ac.ir

(Received May 10, 2016; accepted February 21, 2017)

ABSTRACT

Three-dimensional particulate flow has been simulated using Lattice Boltzmann Method (LBM). Solid-fluid interaction was modeled based on Smoothed Profile Method (SPM) (Jafari *et. al*, Lattice-Boltzmann method combined with smoothed-profile method for particulate suspensions, Phys. Rev. E, 2011). In this paper a GPU code based on three-dimensional lattice Boltzmann method and smoothed profile method has been prepared due to the ability of SPM-LBM to perform locally and in parallel mode. Results obtained for sedimentation of one and two spherical particles as well as their behavior in shear flow showed excellent correspondence with previous published works. Computations for a large number of particles sedimentation showed that combination of LBM and SPM on a GPU platform can be considered as an efficient and promising computational frame work in particulate flow simulations.

Key words: Particulate flow; Parallel processing; Lattice boltzmann method; Smoothed profile method; GPU programming.

NOMENCLATURE

\mathbf{F}	fluid-solid interaction force	f	velocity distribution function
\mathbf{F}_i^H	hydrodynamic force	ε_p	stiffness parameter between particles
\mathbf{F}_i^{ext}	external force	ε_w	stiffness parameter between particles and wall
N_p	number of particle	γ	shear rate
M	velocity transformation matrix	ρ_F	fluid density
m	moments vector	ρ_s	solid density
Re	Reynolds number	τ	relaxation time
R	radius of particle	α	characteristic directions for LBM model
\hat{S}	diagonal relaxation matrix	$\varphi(\mathbf{x}, t)$	density profile
t	time	ν	fluid kinematic viscosity
\bar{u}	fluid velocity vector	ζ	kesi
$u_P(\mathbf{x}, t)$	velocity field of the particles		
V	transitional velocity		

1. INTRODUCTION

Particulate flow occurs in a great number of engineering applications such as fluidized bed, sediment transportation and fuel cell industry.

Analysis of such flow requires accurate prediction of motion of particles, an issue which has its roots in solid-fluid interaction. Although computational fluid dynamic methods has been used in simulation of such flows since more than a few decades ago,

they suffer from intensive computational issues, mainly due to the requirement of moving mesh which is needed to identify particles at subsequent times. As an alternative, Lattice Boltzmann Method (LBM) showed considerable success in simulation of such flows (Ladd and Verberg 2001, Succi 2001), leading to more efficient computer codes capable of predicting fluid-solid interaction in a computationally efficient manner. It is a numerical method for solving Boltzmann equation which is an evolution equation of velocity distribution function in time. This method has three main advantages as compared to other conventional numerical methods, namely simple algorithm, accuracy in satisfying mass and momentum conservation and ideal for parallel processing. Suspensions of particles at low Reynolds number is a remarkable instance of LBM application (Aidun and Clausen 2010).

LBM simulation of particulate flow started by pioneering work of ((Ladd 1994, Ladd 1994) and continued with subsequent works such as (Aidun and Lu 1995) and (Qi 1999). In these studies, momentum exchange method was used to simulate hydrodynamic interaction between solid particles and host fluid based on the bounce-back collision rule which is applied to satisfy the no-slip boundary condition on the fluid-particle interface. In this method, interface is discretized using particle boundary nodes. The main drawback of such method appeared in simulation of particulate flow problems with complex geometry of particles since solid boundary cannot move smoothly in space; hence, it causes fluctuations in the computation of forces exerting on the particle. Although using different techniques such as proper lattice grid with concentrated nodes at boundaries or higher-degree bounce back (Bouzidi, Firdaouss *et al.* 2001) can reduce these fluctuations considerably, it results in increasing computational time .

As an alternative in simulation of solid particles motions in fluid flow, a combination of immersed boundary method (IBM) and LBM was proposed by (Feng and Michaelides 2004). In these methods, a fixed Lagrangian grid was used for particle domains at each time step while a regular Eulerian one was used for fluid domain at all times. Moreover, fluid-solid interaction was implemented through adding an external force density to the fluid domain in order to obtain equal velocity for solid and fluid at the boundary nodes. Transformation of force density from boundary to Eulerian nodes was done by a discrete δ -function which is a complicated task.

In order to avoid the complicated transformation of density force, Smoothed profile method (SPM) was introduced by (Nakayama and Yamamoto 2005) as a simple and efficient method for modeling and simulation of fluid-particle interactions. In SPM, a fixed Eulerian grid is used to simulate fluid flow. It represents the particles by smoothed body forces in the Navier-Stokes equations, instead of treating them as boundary conditions in the fluid. A smoothly spreading interface layer is considered to represent the particle boundaries for transition from the rigid-body motion to the fluid motion. By using

such a simple modification, a regular Cartesian coordinates can be used for a system consisting of many particles with any arbitrary particle shape, rather than boundary-fitted coordinates. Here, solid-fluid interface has a finite volume supported by a few grid points. Thus, simulation of a curved particle in a fixed Cartesian grid can be done without difficulty. Although computational demand of this method depends on the number of grid points, it is insensitive to the number of particles. (Jafari, Yamamoto *et al.* 2011) combined SPM and LBM successfully to study particulate suspension for the first time. In their work, such combination has been validated by simulating flow over a circular cylinder, a neutrally buoyant cylinder in simple shear flow, two circular cylinders that approach each other in a channel, and sedimentation of two circular cylinders in a viscous fluid. All of the results confirmed accuracy and efficiency of the method. Later on (Javaran, Rahnama *et al.* 2013) combined SPM with Lees Edwards boundary condition in order to introduce shear into particle suspensions in two-dimensional flow. Their main reason for using Lees Edward Boundary condition was to simulate a system consisting of large number of particles at a reasonable computational cost and remove wall effects. While Galilean invariance errors were reported in the previous works, their computations showed that such errors were absent in their computations.

Due to the explicit nature of this method and weak correlations among neighbor computational points, LBM is an efficient method for parallel processing especially using Graphic Processing Unit (GPU) (Li, Fan *et al.* , Kuznik, Obrecht *et al.* 2010, Tölke 2010). The use of GPU to perform non-graphical calculations was begun as the computational power of graphic cards starts to increase considerably above that of Central Processing Unit (Anderson, Lorenz *et al.* 2008). As an example, computational power of GeForce 8800-ultra graphic card is 400 GFLOPs in single precision task while that of Intel Core2 Duo 2.4 GHz is 38.4 GFLOPs for the same task (Tölke 2010). These graphic cards have been used by many researchers in their work. (Li, Fan *et al.*) used GPU computations in their 3D LBM code. They obtained a computational time speedup of 15.3 for GPU as compared with CPU. (Kuznik, Obrecht *et al.* 2010) offered a method to implement various parts of the lattice Boltzmann method on GPU. (Bernaschi, Rossi *et al.* 2009) could present a GPU implementation of the multicomponent of LB equation for soft-glassy materials successfully. They obtained the significant performance for their GPU/CPU code speeding up in excess of 10 for 1024^2 grids. A GPU implementation as used by (Bernaschi, Fatica *et al.* 2010) to perform a multi-physics/scale simulation software. They optimized the indirect addressing that a multi-physics/scale simulation software uses for efficient simulations of irregular domains. As Smoothed Profile method, similar to Lattice Boltzmann method, has the ability of performing locally and in parallel mode, their combination can be used in parallel mode by means of GPU in investigating particulate suspensions.

In this paper, multi-relaxation time LBM combined with Smoothed Profile method and GPU computing were used in order to investigate motion of a number of particles in fluid flow. The capability and computational efficiency of the present method are shown by 3D simulation of sedimentation of 1176 spherical particles in a square cylinders in addition to simple geometries such as sedimentation of one and two spherical particles in a quiescent fluid. In the following sections, a brief description of the methods including LBM, SPM and GPU are presented with the subsequent validation tests including particle sedimentations and particles in shear flow. Concluding remarks are given in the last section.

2. COMPUTATIONAL METHOD

The present computational method is based on Lattice Boltzmann Method for simulating fluid flow. LBM is an efficient and simple tool showing remarkable progress in simulating particulate flow during last decade (Aidun and Clausen 2010). While different methods have been used for predicting particle motion in such flows, SPM was selected in the present computations for particle motion simulation due to some advantages which were mentioned in the introduction part of the paper. Because of increasing computational demand required for simulation of large number of particles motion in a fluid flow, attempts were made to use GPU computing. A more detailed explanation of these issues are presented in the following sections.

2.1. Generalized Lattice Boltzmann Method with Forcing Term

LBM is a numerical solution of Boltzmann Equation on a lattice with a discrete velocity field. The main concept in LBM is velocity distribution function which is obtained based on collision and streaming steps. At the collision step, relaxations is considered to the local equilibrium values while movements along the characteristic directions given by a discrete particle velocity space are described in the streaming step. In its simple form, single relaxation time is used in LBM simulation of fluid flows. However, to improve the numerical stability and accuracy of computations, (Yiotis, Psihogios *et al.* 2007) proposed Multiple Relaxation Time (MRT) that can be used in such flow simulations. In MRT-LBM which is also called Generalized LBM, collision is computed at moment space, while streaming step is performed at article velocity space. GLBM has been modified for the case of fluid flows with external forces which is called GLBM with forcing term (Pattison, Premnath *et al.* 2009, Premnath, Pattison *et al.* 2009, Premnath, Pattison *et al.* 2009). It can be express in the following form:

$$f(\vec{x} + \vec{e}_i \delta_t, t + \delta_t) = f(\vec{x}, t) - M^{-1} \cdot \hat{S} [m - m^{eq}(\rho, \vec{u})](\vec{x}, t) + M^{-1} \cdot (I - \frac{1}{2} \hat{S}) \cdot s^f(\vec{x}, t) \quad (1)$$

Here, f is velocity distribution function, M and \hat{S} are the transformation matrix and the diagonal relaxation matrix, respectively. Collision and source terms are expressed in moment space in this equation. M is an orthogonal transformation matrix with 19×19 elements, mapping velocity distribution function to the moment vector m in the moment space. The collision matrix in velocity space, Λ , is related to \hat{S} in Eq. 1 through the relation $\hat{S} = M \Lambda M^{-1}$ such that \hat{S} is a diagonal matrix. The diagonal relaxation matrix is given by:

$$\Lambda = \text{diag}(0, S_\epsilon, S_\epsilon, 0, S_q, 0, S_q, S_\nu, S_\pi, S_\nu, S_\nu, S_\nu, S_m, S_m, S_m) \quad (2)$$

The speed of sound is $c_s = \frac{1}{\sqrt{3}}$ and viscosity is related to S_ν as:

$$\nu = \frac{1}{3} \left(\frac{1}{S_\nu} - \frac{1}{2} \right) \quad (3)$$

Relaxation rate, S_ν , is obtained from Eq. 3 and other relaxation rate $\{S_q, S_\epsilon, S_\nu, S_\pi, S_m\}$ are set by (Sheikh and Pak 2015) to be identical values to preserve symmetry on the chosen lattice as follows:

$$8 \left(\frac{2 - S_\nu}{8 - S_\nu} \right) \quad (4)$$

The last term in Eq. 1 shows the effect of an external force field on the evolution of the distribution function. While different external force fields may exist such as gravity, here fluid-solid interaction force, F which is obtained from SPM method, is considered as an external force field.

The source terms, s^f in moment space are functions of external force and introduced by the following relations:

$$\begin{aligned} s_0^f &= 0 \\ s_1^f &= 38 [F_x u_x + F_y u_y + F_z u_z] \\ s_2^f &= -11 [F_x u_x + F_y u_y + F_z u_z] \\ s_3^f &= [F_x] \\ s_4^f &= -\frac{2}{3} [F_x] \\ s_5^f &= [F_y] \\ s_6^f &= -\frac{2}{3} [F_y] \\ s_7^f &= [F_z] \\ s_8^f &= -\frac{2}{3} [F_y] \\ s_9^f &= 2 [2F_x u_x - F_y u_y - F_z u_z] \end{aligned}$$

$$\begin{aligned}
 S_{10}^f &= -[2F_x u_x - F_y u_y - F_z u_z] \\
 S_{11}^f &= 2[F_y u_y - F_z u_z] \\
 S_{12}^f &= -[F_y u_y - F_z u_z] \\
 S_{13}^f &= [F_x u_y + F_y u_x] \\
 S_{14}^f &= [F_y u_z + F_z u_y] \\
 S_{15}^f &= [F_y u_z - F_z u_x] \\
 S_{16}^f &= S_{17}^f = S_{18}^f = 0
 \end{aligned}
 \tag{5}$$

More detailed on MRT- D_3Q_{19} can be found in Refs. (Mei, Luo *et al.* 2006). In D_3Q_{19} lattice model (Fig. 1), there exist nineteen velocities to resolve the three-dimensional fluid flow. Discrete velocities vector are given as:

$$e = \begin{cases} (0,0,0) & \alpha=0 \\ (\pm 1,0,0), (0,\pm 1,0), (0,0,\pm 1) & \alpha=1-6 \\ (\pm 1,\pm 1,0), (0,\pm 1,\pm 1), (\pm 1,0,\pm 1) & \alpha=7-18 \end{cases}
 \tag{6}$$

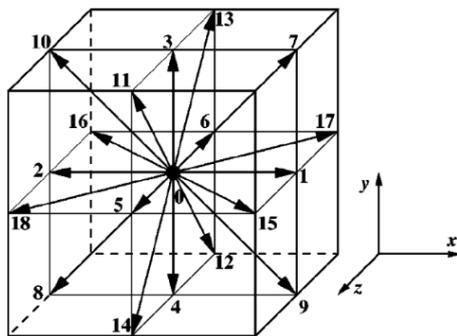


Fig. 1. LBM discrete velocity vector, D_3Q_{19} model.

Finally, the macroscopic density and velocity on each lattice node are given by the following relations:

$$\rho(\mathbf{x}_i, t) = \sum_{\alpha=0}^8 f_{\alpha}(\mathbf{x}_i, t)
 \tag{7}$$

$$\rho(\mathbf{x}_i, t) \mathbf{u}(\mathbf{x}_i, t) = \sum_{\alpha=1}^8 \mathbf{c}_{\alpha} f_{\alpha}(\mathbf{x}_i, t)$$

2.2. Smoothed Profile Method

Smoothed Profile Method is a numerical method used for simulation of fluid-particle interactions. In this method, implementation of no-slip boundary condition is performed using a body force which is introduced in LB equation. The position of solid particles in a fluid are indicated using a function which has a value of one and zero for internal and external nodes respectively, while at the interface, it varies continuously from one to zero as is shown in (Fig. 2).

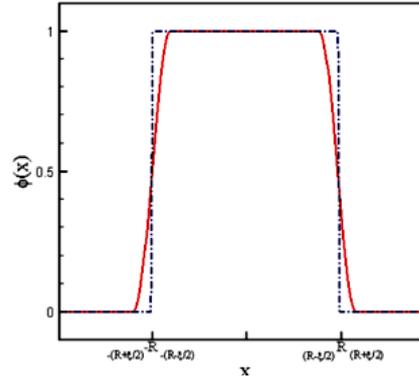


Fig. 2. Representation of a particle in smoothed profile (solid line).

A formulation for such function for N_p particles can be written as:

$$\varphi(\mathbf{x}, t) = \sum_{i=1}^{N_p} \varphi_i(\mathbf{x}, t)
 \tag{8}$$

Where, $\varphi_i(\mathbf{x}, t)$ is the density profile of the i^{th} particle and N_p is the total number of particles in the domain. Several functions have been suggested in the literature of (Nakayama and Yamamoto 2005) for density profile. The function used in the present simulation is expressed as:

$$\begin{aligned}
 \varphi_i(\mathbf{x}, t) &= s(R_i - |\mathbf{x} - \mathbf{R}_i(t)|) \\
 s(x) &= \begin{cases} 0 & x < -\xi/2 \\ \frac{1}{2} \sin\left(\frac{\pi x}{\xi_i} + 1\right) & |x| \leq \xi/2 \\ 1 & x > \xi/2 \end{cases}
 \end{aligned}
 \tag{9}$$

Where R_i is the radius of the i^{th} particle and \mathbf{R}_i is the position vector for center of mass of i^{th} particle.

The velocity field of the particles, $u_p(\mathbf{x}, t)$, is defined as:

$$\begin{aligned}
 \varphi(\mathbf{x}, t) u_p(\mathbf{x}, t) &= \\
 \sum_{i=1}^{N_p} \varphi_i(\mathbf{x}, t) [\mathbf{V}_i(t) + \boldsymbol{\omega}_i \times \{\mathbf{x} - \mathbf{R}_i(t)\}]
 \end{aligned}
 \tag{10}$$

Here, \mathbf{v}_i and $\boldsymbol{\omega}_i$ are the translational and angular velocities of the i^{th} particle, respectively. Fluid nodes covered by the solid particles must have the same velocity as the solid particles. To enforce the fictitious fluid inside the rigid particle to satisfy the rigid-body motion constraint, a body force is introduced inside of the particle domain which is zero outside of it. Fluid-solid interaction force which acts on solid particle nodes is given by (Jafari, Yamamoto *et al.* 2011):

$$\begin{aligned}
 \varphi(\mathbf{x}, t) f_P(\mathbf{x}, t) &= \\
 \varphi(\mathbf{x}, t) (u_P(\mathbf{x}, t) - u(\mathbf{x}, t)) / \delta t
 \end{aligned}
 \tag{11}$$

In the above equation, $u_p(\mathbf{x}, t)$ and $u(\mathbf{x}, t)$ are the

particle and fluid velocities at t and \mathbf{x} , respectively. The force applying on the fluid boundary nodes is obtained from the following relation:

$$\mathbf{f}_H(\mathbf{x}, t) = -\varphi(\mathbf{x}, t)\mathbf{f}_p(\mathbf{x}, t) = -\varphi\left(u_p(\mathbf{x}, t) - u(\mathbf{x}, t)\right) / \delta t \quad (12)$$

In order to implement interaction between two phases, fluid-solid interaction force is added in Eq. 1. Applying momentum conservation law, hydrodynamic force and torque on the particles, can be obtained from the following relations:

$$\mathbf{F}_i^H = \int_{V_{p_i}} \rho\varphi\left(u(\mathbf{x}, t) - u_p(\mathbf{x}, t)\right) dV_{p_i} \quad (13)$$

$$\mathbf{T}_i^H = \int_{V_{p_i}} (\mathbf{x} - \mathbf{R}_i) \times \rho\varphi\left(u(\mathbf{x}, t) - u_p(\mathbf{x}, t)\right) dV_{p_i} \quad (14)$$

Finally, particle translational and angular velocities and new particle position are obtained using the following equations:

$$\mathbf{V}_i^{n+1} = \mathbf{V}_i^n + M_{p_i}^{-1} \int_{t_n}^{t_n + \delta t} \left(\mathbf{F}_i^H + \mathbf{F}_i^{\text{lb}} + \mathbf{F}_i^{\text{ext}}\right) ds \quad (15)$$

$$\boldsymbol{\omega}_i^{n+1} = \boldsymbol{\omega}_i^n + \mathbf{I}_{p_i}^{-1} \int_{t_n}^{t_n + \delta t} \left(\mathbf{T}_i^H + \mathbf{T}_i^{\text{ext}}\right) ds \quad (16)$$

$$\mathbf{R}_i^{n+1} = \mathbf{R}_i^n + \int_{t_n}^{t_n + \delta t} \mathbf{V}_i ds \quad (17)$$

In which, $\mathbf{F}_i^{\text{ext}}$, $\mathbf{T}_i^{\text{ext}}$ are the external force and torque on particle and \mathbf{F}_i^{lb} is the collision force between two particles or a particle and a wall in distances of the order of grid spacing. Here M_{p_i} and \mathbf{I}_{p_i} are the mass and the moment of inertia tensor for the i^{th} particle, respectively. They are obtained from the following relations for a sphere:

$$M_{p_i} = \frac{4}{3} \rho_s \Pi R^3 \quad (18)$$

$$\mathbf{I}_{p_i} = 0.4 M_{p_i} R^2 \quad (19)$$

Where ρ_s is the density of the i^{th} particle. Detailed derivation of the equations mentioned in this section were presented in (Nakayama and Yamamoto 2005); (Jafari, Yamamoto *et al.* 2011).

2.3. Particles Near Contact

In fluid flows containing many particles, collision between particles is unavoidable. Generally, forces between particles can be modeled through fluid-solid interaction force when the distance between the nearest points on the surfaces of two particles contains enough fluid-computational nodes. For the case of small distances in the order of one lattice spacing, no fluid-solid interaction force can be computed due to the lack of fluid nodes. This situation results in particles crossing each other. Therefore an artificial repelling force is needed to prevent particles from penetrating into each other or

into domain walls when the distance between two adjacent solid nodes become less than one grid distance. A popular approach is to introduce a repulsive force when the gap between two particles exceeds a given threshold, the so-called safe zone. In this study, a pair-wise repulsive force acting on the i^{th} particle due to its interaction with j^{th} particle is adopted. This extra short-range repulsive force which is added as an external force into the total force experienced by a particle, has a functional form as follows (Feng and Michaelides 2004):

$$\mathbf{F}_{ij}^P = \begin{cases} 0 & |\mathbf{R}_i - \mathbf{R}_j| > 2R_p + \zeta \\ c_{ij} \left(\frac{|\mathbf{R}_i - \mathbf{R}_j| - 2R_p - \zeta}{\zeta} \right)^2 \left(\frac{\mathbf{R}_i - \mathbf{R}_j}{|\mathbf{R}_i - \mathbf{R}_j|} \right) & |\mathbf{R}_i - \mathbf{R}_j| \leq 2R_p + \zeta \end{cases} \quad (20)$$

Similarly the repulsive force between a particle and a wall is given by a reflection method which is expressed as follows:

$$\mathbf{F}_{ij}^W = \begin{cases} 0 & |\mathbf{R}_i - \mathbf{R}_{wj}| > 2R_p + \zeta \\ c_{ij} \left(\frac{|\mathbf{R}_i - \mathbf{R}_{wj}| - 2R_p - \zeta}{\zeta} \right)^2 \left(\frac{\mathbf{R}_i - \mathbf{R}_{wj}}{|\mathbf{R}_i - \mathbf{R}_{wj}|} \right) & |\mathbf{R}_i - \mathbf{R}_{wj}| \leq 2R_p + \zeta \end{cases} \quad (21)$$

In the above two equations, c_{ij} is a force scale and is set to be the buoyancy force in this study. \mathbf{R}_i and \mathbf{R}_j represent particle positions, \mathbf{R}_{wj} is wall position and ζ is the threshold or safe zone and is set to one lattice unit in the present simulation. ε_p and ε_w are the stiffness parameters. This repulsive force is called lubrication force in most published papers.

3. PARALLEL PROCESSING USING GPU

Nowadays, a Graphics Processing Unit (GPU) is widely used to perform a very large number of same or similar calculations on data items very quickly. GPUs are a computing hardware with a large collection of only modestly powerful cores (1536 on the NVidia GeForce GTX 680) as compared to a CPU which is a small collection of very powerful cores (Intel® Xeon® CPU @ 2.4 GHz). To run a program on GeForce graphic cards, NVidia CUDA C programming language was introduced and a code should be prepared as a CUDA C code.

CUDA is a programming language based on the standard C language for parallel processing on graphic cards. The structure of a CUDA program reflects the cooperation of CPU and GPU. As an advantage of LBM/SPM, their local computations can be used in parallel simulation of particulate flows. These local calculation are explicit and generally needs only the information from nearest adjacent nodes. In LBM, macroscopic flow properties are obtained based on the motions of constituent particles. As calculation of each particle is performed independently, they can be done

simultaneously; consequently, GPU computing can be used as a means of parallel computation. The first step in a GPU code is to transfer data from CPU to the global memory of GPU. When the information were placed on the GPU, each data grid in the physical geometry is broken down into block threads and threads. Each grid point of the physical geometry is connected to its respective thread. A thread number identifies each thread. This thread number is obtained according to its row, column and depth. The physical grid versus the computing thread blocks and threads is illustrated in (Fig. 3). Due to the large number of threads exist in GPU, it is possible to do parallel computations on each grid at the same time which results in considerable reduction of computational time. The present computations were performed using a GPU code which was prepared by authors.

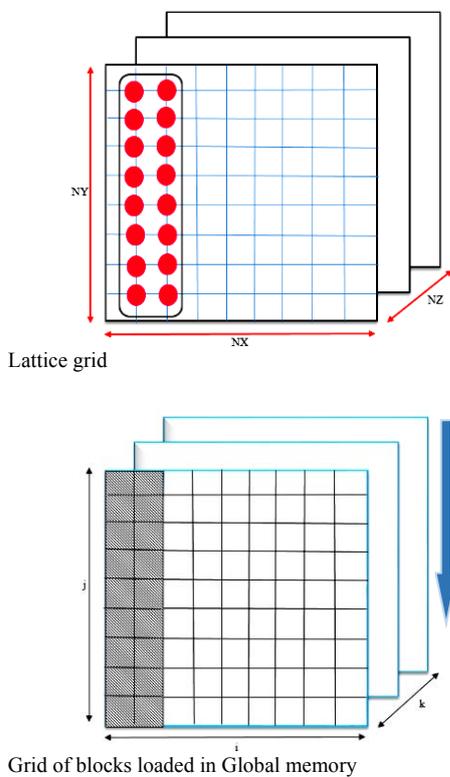


Fig. 3. Physical grid and thread blocks.

4. VALIDATION OF NUMERICAL RESULTS

In order to validate the present computational method, a series of flow geometries which were studied by previous researchers are modeled and compared with their published work. These are sedimentation of one and two spheres in a quiescent fluid as well as their motion in shear flow. In addition to these flow geometries, sedimentation of 1176 particles in a quiescent fluid were studied to reveal the computational ability of the present method. Such simulation was also performed in the present work which is explained in the following

sections.

4.1. A Particle Sedimentation in a Quiescent Fluid

Sedimentation of a sphere in a square cylinder containing quiescent fluid has been investigated by many researchers. (Ten Cate, Nieuwstad *et al.* 2002) used particle image velocimetry to study this geometry. In their experiments, dimensions of the square cylinder was selected as $100 \times 100 \times 160 \text{ mm}^3$, as is shown in (Fig. 4). A sphere with a diameter of 15 mm was located at center at a height of 120 mm from bottom of the cylinder. Density of sphere was selected as 1120 kg/m^3 , while different quiescent fluid were used to obtain different Reynolds number. (Table 1) shows selected fluid viscosities which are correspond to $Re = 1.4, 4.1, 11.6$ and 31.9 . Last column of table shows sphere terminal velocity after its release from rest.

Table 1 Reynolds number, fluid density and viscosity and terminal velocity of sphere in its sedimentation. (Re , ρ_F and μ_F were taken from Ten cate *et al.*(2002))

$Re_p = \frac{\rho_F U_\infty d_p}{\mu_F}$	$\rho_F \left(\frac{\text{kg}}{\text{m}^3} \right)$	$\mu_F \left(\frac{10^{-3} \text{Ns}}{\text{m}^2} \right)$	$U_\infty \left(\frac{\text{m}}{\text{s}} \right)$
1.5	970	373	0.038
4.1	965	212	0.06
11.6	962	113	0.09
31.9	960	58	0.128

Spherical particle sedimentation was studied using LBM-SPM-GPU code in the present work. The number of grid points used in this simulation were $100 \times 100 \times 160$. Using this number of grid points for the aforementioned geometry results in sphere diameter 15 based on lattice units. Relaxation time was selected as 0.8. Physically, the sphere starts to accelerate after its release, then it attains a constant terminal velocity which occurs due to balance of gravitational and drag forces. Near bottom of cylinder, as sphere approaches the bottom, its velocity decreases until it touches the bottom with zero velocity.

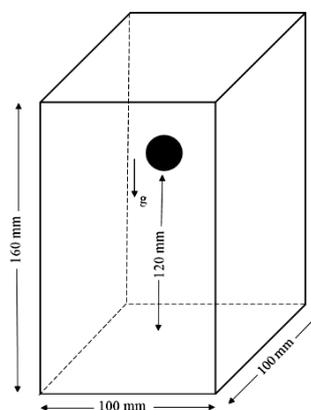


Fig. 4. Schematic view of a sphere in a square cylinder.

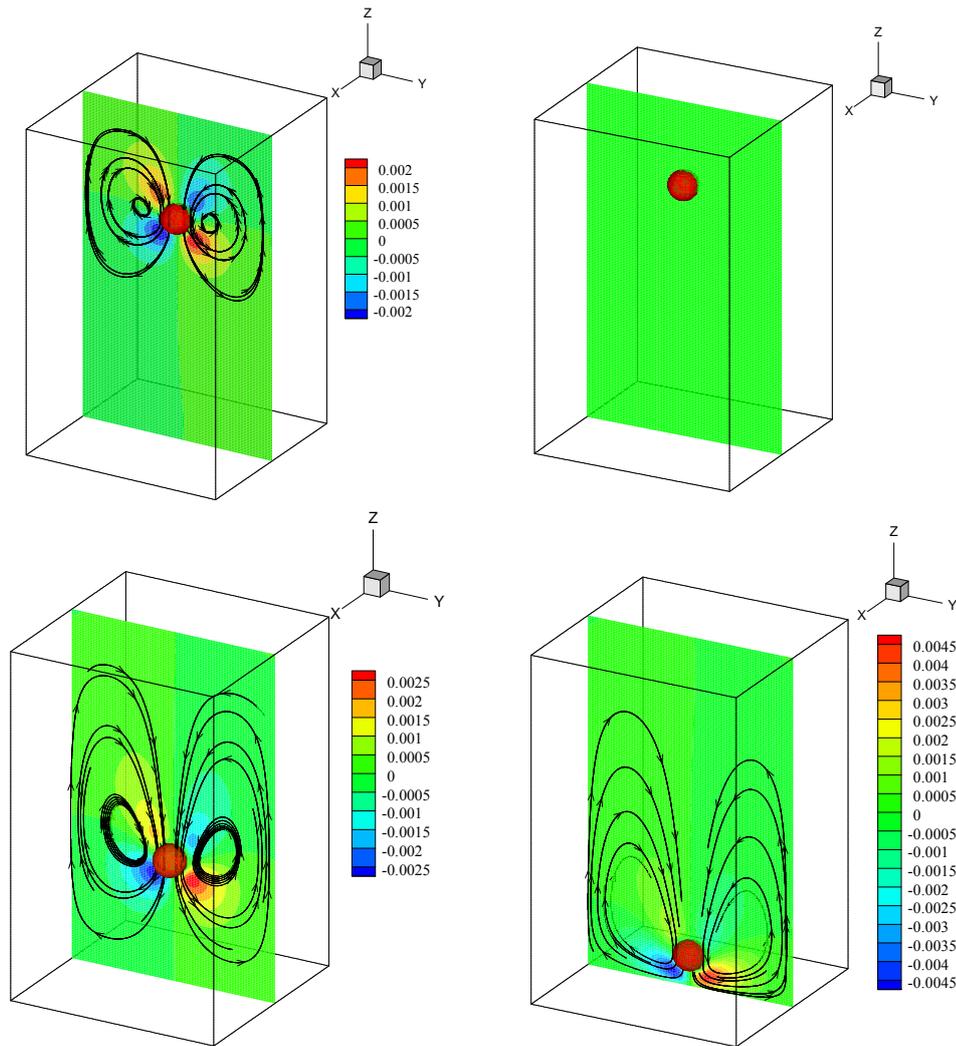


Fig. 5. Vertical velocity contours with streamlines for falling sphere at four instances.

(Figure 5) represents vertical velocity contours of the fluid with streamlines configuration during sedimentation of the sphere at different times. A symmetrical behavior relative to the vertical axis is observed in this figure. (Fig. 6) shows particle settling velocity and location of center of sphere as a function of time along with experimental results obtained by (Ten Cate, Nieuwstad *et al.* 2002) at different Reynolds numbers. As is observed in (Fig. 6(a)), vertical velocity obtained from present computations shows acceptable correspondence to those obtained experimentally by (Ten Cate, Nieuwstad *et al.* 2002). (Fig. 6(b)) shows good agreement between experimental results and the present computations considering spatial position of sphere with time. In general, it can be concluded that the present computations are capable of predicting single particle sedimentation with good accuracy.

To reveal the effect of number of grid points used in the present computations, effect of grid resolution on sedimentation velocity at $Re=11.6$ is studied by increasing the number of grid points from

$64 \times 64 \times 104$ to $130 \times 130 \times 208$ as is shown in (Fig. 7). As observed in this figure, increasing the number of grid points does not improve sedimentation velocity considerably compared to experimental data.

4. 2. Sedimentation of two in-line Spheres in a Quiescent Fluid

When two in-line spheres settle under gravity in a quiescent fluid, their position does not follow a regular pattern due to their wake interactions. This problem has been investigated by (Glowinski, Pan *et al.* 2001) and (Apte, Martin *et al.* 2009). (Fig. 8) shows geometry of two spheres in a cylinder before starting to move. The computational domain considered for this problem was $10 \times 10 \times 40$ mm³. Densities of each of two identical particles and the surrounding fluid are 1.14 g/cm³ and 1.0 g/cm³, respectively. Fluid viscosity was selected as 0.01 g/cm.s. Initially, two particles, each of diameter 1/6 cm, are placed in-line in the fluid at the center of square cylinder at different heights of 3.5 cm and 3.16 cm. Both particles start to settle from rest

under gravity. The number of grid points used in the present geometry were selected as $96 \times 96 \times 384$. A relaxation time of 0.6 was implemented for the present calculation.

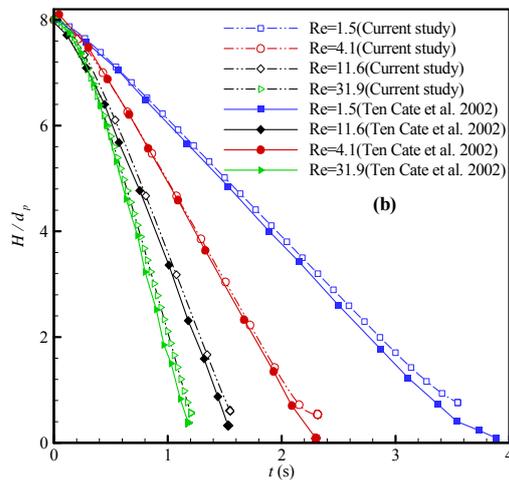
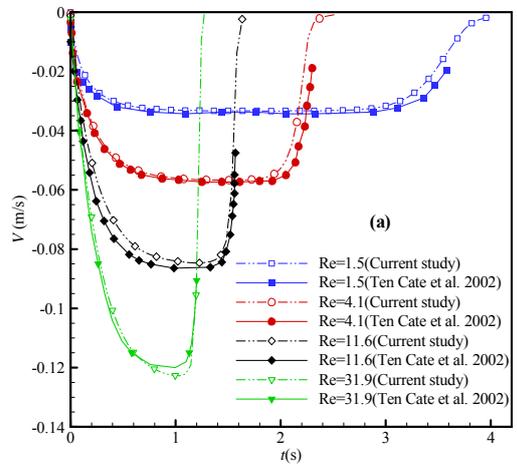


Fig. 6. Variation of vertical (a) velocity and (b) height of sphere with time for different Reynolds numbers.

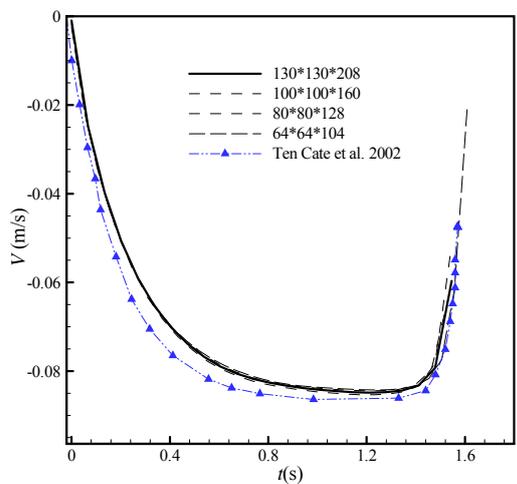


Fig. 7. Comparison of particle settling velocity for $Re=11.6$ for different grid size.

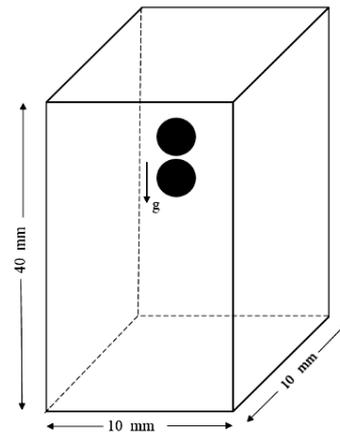


Fig. 8. Schematic presentation of the geometry of two spheres placed in a square cylinder.

Figures 9 and 10 show separation distance between centers of spheres and vertical position as a function of time respectively. It is observed that distance between two spheres remains constant until $t=0.17$ sec. This distance decreases until $t=0.32$ sec., the time which two spheres are nearly in touch with each other. This period of motion is called "drafting". From $t=0.32$ to $t=0.57$, the spheres remain nearly in touch with each other, a period which is called "kissing". It is worth mentioning that any numerical simulation used for prediction of motion during kissing period, should be able to maintain the sphere in touch with each other, while preventing any penetration of particles into each other. Implementation of a repulsive artificial force called lubrication force, prevents spheres from intersecting each other. More detailed discussion about lubrication force was mentioned in section 2.3. In the last period of motion which is called "tumbling", the distance between them increases which is due to the deceleration of spheres as well as their lateral motion. As is observed in fig. 9 and 10, the present results correspond to the results that were obtained from other researcher such as ((Glowinski, Pan *et al.* 2001);(Apte, Martin *et al.* 2009).

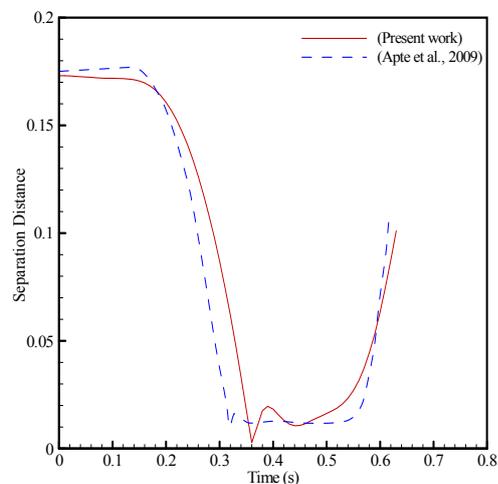


Fig. 9. Separation distance between two spheres settling in a quiescent fluid.

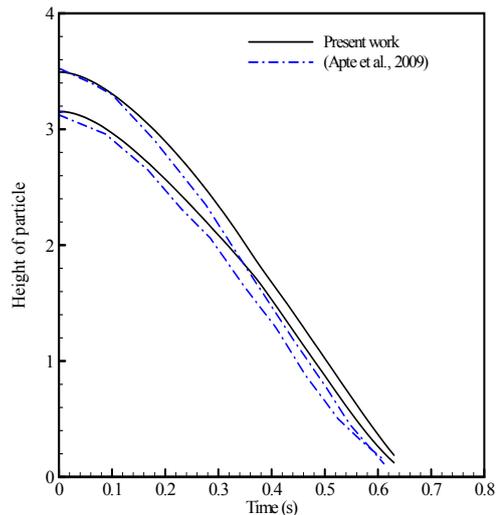


Fig. 10. Vertical position of two sphere setting in a quiescent fluid.

Vertical velocity contour of fluid with streamlines during sedimentation of the two spheres at different times represent by (Fig. 11).

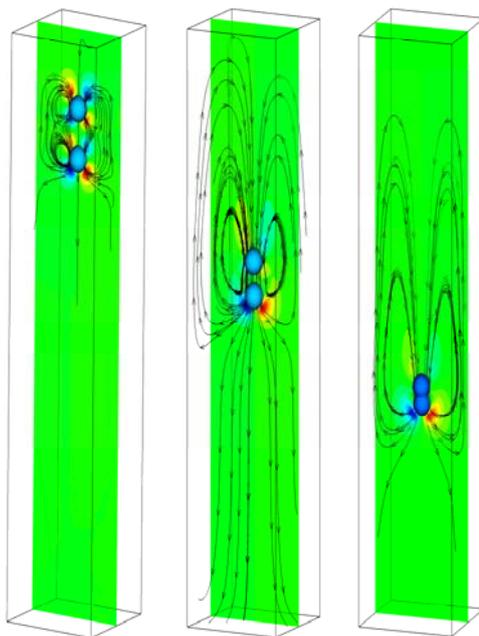


Fig. 11. Two settling particles in a quiescent flow with streamline configuration and vertical velocity contour.

4. 3. Sedimentation of Many Spheres in a Quiescent Fluid

4.3.1. The Sedimentation Process

In this section, sedimentation of 1176 spheres in a quiescent fluid were studied to reveal the

ability of present LBM-SPM in simulation of such flow geometries. All spheres have the same diameter of 0.06 cm which is equal to 6 based on lattice units. Initially, they were placed at top of a cubic enclosure of length 1.00 cm, see fig. 12a. Initial gap between two neighboring particles is one lattice unit. Particle and fluid densities are 1.04 g/cm^3 and 1 g/cm^3 respectively. Particles position at subsequent times are presented in (Figs.12a-d). (Fig. 12a) shows initial position of particles, which are placed in six rows starting from top of the enclosure. The first row at the top is placed at half lattice unit from top wall. No-slip boundary is adopted for all boundaries. Figs. 12b-12d show particle positions at subsequent times after starting to settle under gravity. As is observed from these figures, particles located at middle settle down more quickly than those placed near the walls. Fig. 13 shows variation of average velocity of particles located at the top row as well as those at the bottom row with time. As is observed from these figures, top row achieves higher vertical downward velocity after about 3000 time steps compared to the bottom row. This is due to the wake space appeared after displacement of other rows making the first row to move faster. It was also observed that bottom row is nearly settled at the bottom of the computational domain after about 5000 time steps. Such three-dimensional behavior is almost opposite to two-dimensional results reported by previous authors such as (Feng, Hu *et al.* 1994, Feng and Michaelides 2004). (Zhang, Trias *et al.* 2015) who studied three-dimensional particle sedimentation, reported the same behavior, which confirms our findings in such flow geometry.

4.3.2. Effect of GPU on Computational Time

In order to examine the effect of GPU computing on computational time as compared to CPU one, several simulations were done for different number of particles. Particles were arranged in the same region as previous subsection. The diameter of particles is 0.06 cm. Computational time for one iteration as a function of number of particles presented in (Fig. 14). As is observed in this figure, computational time increases almost linearly with increasing the number of particles for both GPU and CPU computations, however, higher slope is observed for CPU compared to GPU computations. Computational time for simulation of sedimentation of 3060 particles using CPU is 5 times that of GPU which confirms computational efficiency of LBM/SPM with GPU in particulate flow simulations.

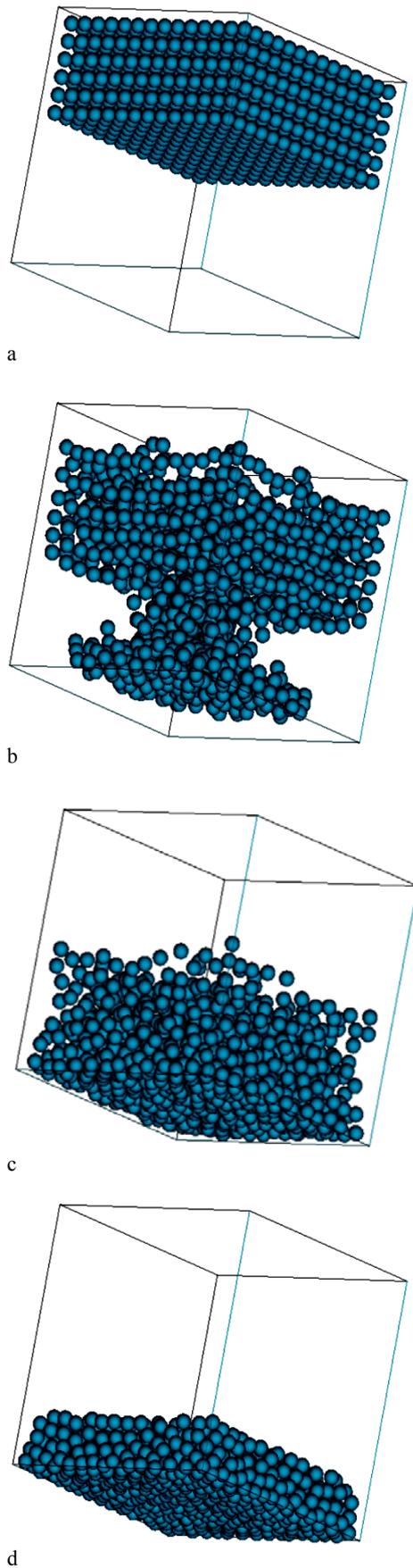


Fig. 12. Positions of the 1176 particles at different times, a= 0.0 s, b=5.7, c=11s, d=20 s.

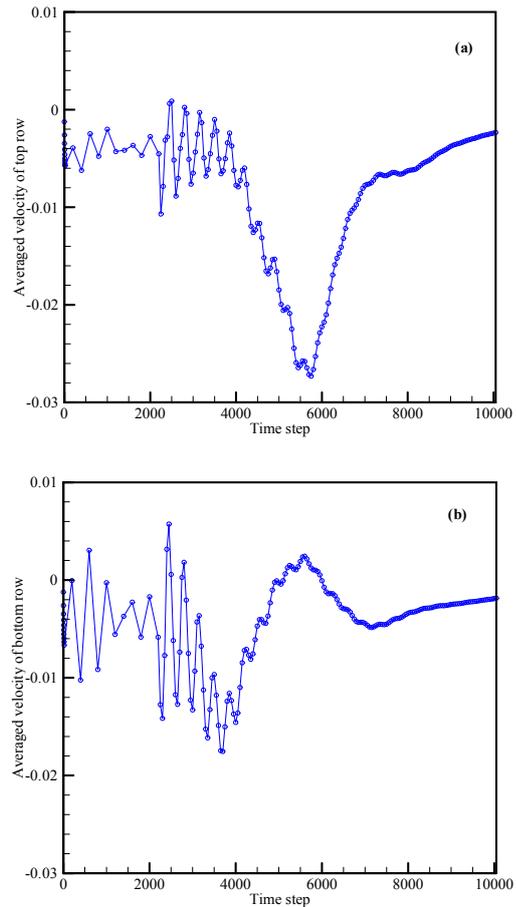


Fig. 13. Average velocity of sedimentation of particles located at (a) the top and (b) the bottom row

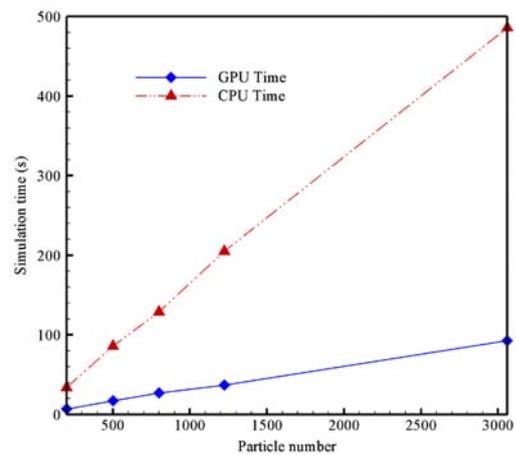


Fig. 14. Particle number versus simulation time.

4.4. Spheres in Shear Flow

To reveal the behavior of present computational scheme in shear flows, the problem of motion of a sphere located at centerline of a shear flow was studied. (Fig. 15) shows schematic of a sphere located midway between two parallel plates. As is shown in it, the top and bottom plates are moving in opposite directions, creating a shear flow which causes sphere to rotate. Computational domain

consists of two solid walls for which bounce back boundary conditions are applied, with periodic boundary conditions on remaining boundaries. Density of solid sphere is selected as 10 times that of fluid. Reynolds number based on sphere diameter is defined as $Re_p = 4\gamma R_p^2 / \nu$ in which γ is shear rate and defined as $\gamma = 2U_w / L_y$.

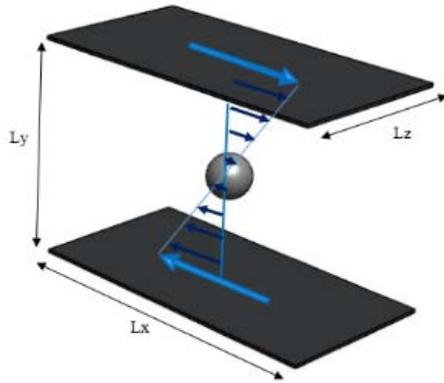


Fig. 15. Schematic of the geometry of a sphere located between two parallel plates in a shear flow.

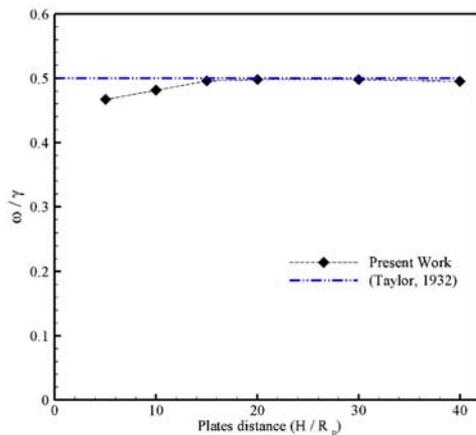


Fig.16. Effect of plate distance on angular non-dimensional velocity of sphere. The results are compared to results of (Taylor 1932) for a 3D system.

Non-dimensional angular velocity of the sphere obtained from exact solution is equal to 0.5 in this geometry (Taylor 1932). To investigate the effect of distance between two plates on angular velocity of sphere, it was changed from $2.5R_p$ to $40R_p$. (Fig. 16) shows the effect of plates separation distance on non-dimensional angular velocity of the sphere at $Re_p = 0.058$. As it is observed in this figure, the predicted angular velocity is equal to the exact solution if the distance between two plates is selected greater than $16R_p$. The discrepancy observed in fig. 16 occurs if the distance between plates is less than $16R_p$. The reason for such discrepancy is the wall effects which appear due to higher shear rate related to smaller distance. As the

vertical axis of fig. 16 shows non-dimensional angular velocity, increasing shear rate reduces this parameter resulting in values lower than 0.5. It is worth mentioning that reduction in angular velocity with increasing wall distances beyond $16R_p$ is proportional to the reduction in shear rate, resulting in a constant non-dimensional angular velocity.

In order to study the hydrodynamic interactions between two particles using the combined SPM-LBM, the problem of two spheres located in a shear flow is investigated. When two spherical particles which are located at a distance from each other, are subjected to shear flow, they start to approach each other. (Fig. 17) line of the channel. Consider a coordinate axis with origin located at centerline, midway between inlet and outlet, particles are located at $(5R_p, -0.75R_p, 0)$ and $(-5R_p, 0.75R_p, 0)$. The plate separation distance affects angular and translational velocity of spheres. Results of the present computations showed that no significant variation in these two parameters is observed for $H > 40R_p$. Therefore distance between two plates was selected as $H = 40R_p$.

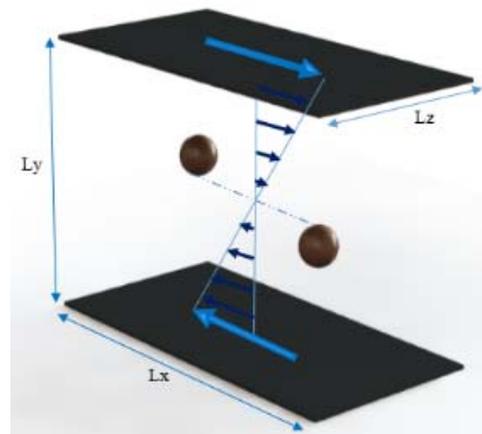


Fig. 17. Schematic of two spherical particles in a shear flow.

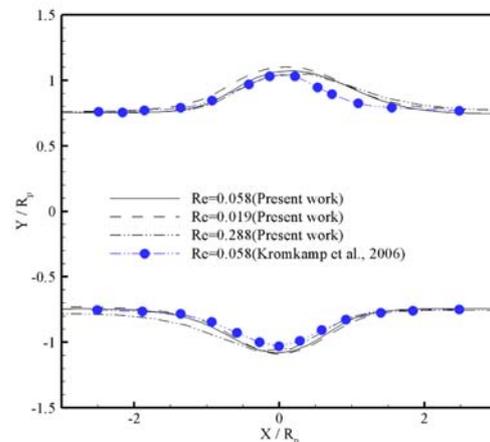


Fig. 18. Central position of two spheres in shear flow as a function of time.

Figure 18 shows location of two spherical particles as a function of time along with those obtained by

(Kromkamp, van den Ende *et al.* 2006). When two spheres approach each other, they experience a vertical motion in opposite direction to be able to pass each other. As is observed in this figure, the present results show good correspondence with those of (Kromkamp, van den Ende *et al.* 2006).

5. CONCLUSION

A computational frame work consisting of Lattice Boltzmann Method and Smoothed Profile Method which is based on GPU computations, is used to study particulate flow. Results obtained for one and two particle sedimentations in a quiescent flow showed accurate prediction of particle position as well as its velocity during sedimentation as compared to previous published works. Incorporation of GPU computations made it possible to simulate sedimentation of 1176 particles in a reasonable computational time as well as the ability of the method in studying a large number of particle motions in a fluid flow. To reveal the ability of present method in predicting particle motion in shear flow, flow geometries consisting of one and two spherical particles in shear flow were studied in detail. Results showed accurate prediction of particle angular velocity for the case of one particle in shear flow, as well as motion of two particles passing each other. In general, the present computations showed that LBM/SPM with GPU computing is capable of predicting accurate particulate flow characteristics with high computational efficiency.

REFERENCE

- Aidun, C. K. and J. R. Clausen (2010). Lattice-Boltzmann method for complex flows. *Annual review of fluid mechanics* 42, 439-472.
- Aidun, C. K. and Y. Lu (1995). Lattice Boltzmann simulation of solid particles suspended in fluid. *Journal of Statistical Physics* 81(1-2), 49-61.
- Anderson, J. A., C. D. Lorenz and A. Travesset (2008). General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of computational physics* 227(10), 5342-5359.
- Apte, S. V., M. Martin and N. A. Patankar (2009). A numerical method for fully resolved simulation (FRS) of rigid particle-flow interactions in complex flows. *Journal of computational physics* 228(8), 2712-2738.
- Bernaschi, M., L. Rossi, R. Benzi, M. Sbragaglia and S. Succi (2009). Graphics processing unit implementation of lattice Boltzmann models for flowing soft systems. *Physical Review E* 80(6), 066707.
- Bernaschi, M., M. Fatica, S. Melchionna, S. Succi and E. Kaxiras (2010). A flexible high-performance Lattice Boltzmann GPU code for the simulations of fluid flows in complex geometries. *Concurrency and Computation: Practice and Experience* 22(1), 1-14.
- Bouzidi, M. h., M. Firdaouss and P. Lallemand (2001). Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids (1994-present)* 13(11), 3452-3459.
- Feng, J., H. Hu and D. Joseph (1994). Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 2. Couette and Poiseuille flows. *Journal of Fluid Mechanics* 277(271), 271-301.
- Feng, Z. G. and E. E. Michaelides (2004). The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. *Journal of computational physics* 195(2), 602-628.
- Glowinski, R., T. Pan, T. Hesla, D. Joseph and J. Periaux (2001). A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *Journal of computational physics* 169(2), 363-426.
- Jafari, S., R. Yamamoto and M. Rahnama (2011). Lattice-Boltzmann method combined with smoothed-profile method for particulate suspensions. *Physical Review E* 83(2), 026702.
- Javaran, E. J., M. Rahnama and S. Jafari (2013). Combining Lees-Edwards boundary conditions with smoothed profile-lattice Boltzmann methods to introduce shear into particle suspensions. *Advanced Powder Technology* 24(6), 1109-1118.
- Kromkamp, J., D. van den Ende, D. Kandhai, R. van der Sman and R. Boom (2006). Lattice Boltzmann simulation of 2D and 3D non-Brownian suspensions in Couette flow. *Chemical engineering science* 61(2), 858-873.
- Kuznik, F., C. Obrecht, G. Rusauoen and J. J. Roux (2010). LBM based flow simulation using GPU computing processor. *Computers and Mathematics with Applications* 59(7), 2380-2392.
- Ladd, A. and R. Verberg (2001). Lattice-Boltzmann simulations of particle-fluid suspensions. *Journal of Statistical Physics* 104(5-6), 1191-1251.
- Ladd, A. J. (1994). Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *Journal of Fluid Mechanics* 271, 285-309.
- Ladd, A. J. (1994). Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results. *Journal of Fluid Mechanics* 271, 311-339.
- Li, W., Z. Fan, X. Wei and A. Kaufman GPU-based flow simulation with complex boundaries. 2005. *Computer Science Department, SUNY at Stony Brook*.

- Mei, R., L. S. Luo, P. Lallemand and D. d'Humières (2006). Consistent initial conditions for lattice Boltzmann simulations. *Computers & Fluids* 35(8), 855-862.
- Nakayama, Y. and R. Yamamoto (2005). Simulation method to resolve hydrodynamic interactions in colloidal dispersions. *Physical Review E* 71(3), 036707.
- Pattison, M. J., K. N. Premnath and S. Banerjee (2009). Computation of turbulent flow and secondary motions in a square duct using a forced generalized lattice Boltzmann equation. *Physical Review E* 79(2), 026704.
- Premnath, K. N., M. J. Pattison and S. Banerjee (2009). Dynamic subgrid scale modeling of turbulent flows using lattice-Boltzmann method. *Physica A: Statistical Mechanics and its Applications* 388(13), 2640-2658.
- Premnath, K. N., M. J. Pattison and S. Banerjee (2009). Generalized lattice Boltzmann equation with forcing term for computation of wall-bounded turbulent flows. *Physical Review E* 79(2), 026703.
- Qi, D. (1999). Lattice-Boltzmann simulations of particles in non-zero-Reynolds-number flows. *Journal of Fluid Mechanics* 385, 41-62.
- Sheikh, B. and A. Pak (2015). Numerical investigation of the effects of porosity and tortuosity on soil permeability using coupled three-dimensional discrete-element method and lattice Boltzmann method. *Physical Review E* 91(5), 053301.
- Succi, S. (2001). The lattice Boltzmann equation: for fluid dynamics and beyond, Oxford university press.
- Taylor, G. I. (1932). The viscosity of a fluid containing small drops of another fluid. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 41-48.
- Ten Cate, A., C. Nieuwstadt, J. Derksen and H. Van den Akker (2002). Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity. *Physics of Fluids (1994-present)* 14(11), 4012-4025.
- Tölke, J. (2010). Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA. *Computing and Visualization in Science* 13(1), 29-39.
- Yiotis, A. G., J. Psihogios, M. E. Kainourgiakis, A. Papaioannou and A. K. Stubos (2007). A lattice Boltzmann study of viscous coupling effects in immiscible two-phase flow in porous media. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* 300(1), 35-49.
- Zhang, H., F. X. Trias, A. Oliva, D. Yang, Y. Tan, S. Shu and Y. Sheng (2015). PIBM: Particulate immersed boundary method for fluid-particle interaction problems. *Powder Technology* 272, 1-13.