

A multigrid finite element fictitious boundary method for fluid-solid two-phase flows

Khuram Walayat^{1,2}, Zekun Wang^{1,2}, Kamran Usman³ and †Moubin Liu^{1,2}

¹BIC-ESAT, College of Engineering, Peking University, Beijing 100187, China

²State Key Laboratory for Turbulence and Complex Systems, Peking University, Beijing 100871, China

³Department of Mathematics, Faculty of Basic and Applied Sciences, Air University, Islamabad, 44000, Pakistan

†Corresponding author: mbliu@pku.edu.cn

Abstract

This paper presents a direct numerical simulation technique multigrid - finite element fictitious boundary method (FEM-FBM) for the simulation of incompressible viscous fluid flow along with moving rigid bodies. Multigrid finite element solver is used to compute fluid flow on a fixed Eulerian mesh which is independent of particles shape, size and number and the solid particles are allowed to move freely in the whole computational domain. Fictitious boundary method (FBM) is used to treat the particles inside the fluid which takes account the interaction between the fluid and the particles. FBM is based on an explicit volume integral based calculation for the hydrodynamic forces. Particle-particle and particle-wall interactions are handled by collision model proposed by P. Singh et al. [1]. The accuracy and efficiency of presented method is analyzed by four test cases and then numerical tests are performed to show that this method is potentially powerful and provides an efficient approach to simulate complex particulate flows with large number of particles.

Keywords: Direct numerical simulation, Fictitious boundary method, Multigrid, Particulate flows, sedimentation.

Introduction

Particulate flows or motion of solid particles in fluids have a wide range of industrial applications, such as fluidized suspensions, lubricated transport, sedimentation, hydraulic fracturing of reservoirs, slurry flow, paper pulp, food products etc. These types of flows are common in many natural processes such as sand or dust particles in air blown by wind, ocean current interaction with offshore structures, lava flow and sedimentation in estuary etc. Particulate flows in biological processes have been a subject of great importance with research contributions coming from the field of biology, chemistry, physics, engineering and mathematics. Here we are particularly interested in gravitational settling of particles. The sedimentation of suspended particles have a great importance in the chemical, petroleum, paper pulp, wastewater, food, pharmaceutical, ceramic and other industries as a way of separating particles from fluid as well as separating particles of different types settling with different speeds.

Particulate flows are quite complex and hard to simulate numerically, because frequent generation and deformation of computational grid is required in many cases when the particle boundaries are complex and moving with time. The problem becomes more complex in the case with large number of particles due to fluid particle interaction as well as due to particle-particle and particle-wall collisions. Rapid advancements in computational power makes the direct numerical simulation (DNS), an important and practical tool to study particulate flow mechanism. It treats the fluid and solid objects separately. The DNS approach is based on Navier–Stokes equation for the fluid and equations of rigid body motion for particles. A variety of DNS numerical schemes have been proposed over the past decade to simulate fluid-particle flow problems. These methods are broadly classified into two types. One is based on the Lagrangian

approach while the other is the Eulerian approach. In Lagrangian approach, the mesh moves and follows the moving boundaries of the particles in the fluid. Since the motion of the mesh can be defined arbitrarily within the fluid, therefore this approach is usually called Arbitrary Lagrangian Eulerian (ALE). Hu et al. [2, 3], Maury [4, 5] and Feng et al. [6] have used the ALE method to study particulate flows. ALE method normally requires to generate a new mesh at every time step in the case of moving particles, so it is computationally expensive especially for the simulations of problems with large number of particles. Whereas, Eulerian approach is more efficient than the Lagrangian one. Eulerian methods do not require re-meshing, a fixed Cartesian mesh is required which covers the whole computational domain comprising of both particles and fluid. Peskin[7] introduced immersed boundary method (IBM), based on Eulerian approach to study fluid solid interaction problems. Similar to IBM, Glowinski et al. [8-10] developed a finite element fictitious domain method to simulate fluid particle flow problems. Turek et al. [11-13] presented a multigrid finite element fictitious boundary method FBM for the simulation of particulate flows.

In the present work, we use the multigrid FEM fictitious boundary method[11-14] to simulate particle sedimentation problems. The considerable advantage of multigrid FEM fictitious boundary method is that it is based on a fixed FEM background grid which is independent of flow features, hence re-meshing is not required. By applying boundary conditions at the interface between fluid and particles which become an additional constraint to the governing equations, so the fluid domain can be extended into the whole domain which covers both fluid and particles.

Mathematical Modeling

Consider the flow of N number of solid particles of mass $M_i (i = 1, 2, \dots, N)$ in a fluid. The density of the fluid is ρ_f and its viscosity is ν . $\Omega_f(t)$ and $\Omega_i(t)$ denotes the domain occupied by the fluid and the i^{th} particle at time t respectively.

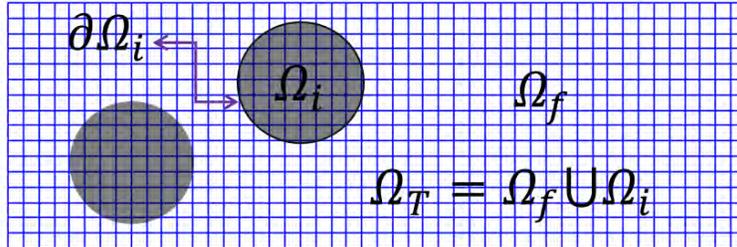


Figure 1. Rigid particle and fluid

whereas Ω_T is the total domain and is given by

$$\Omega_T = \Omega_f(t) \cup \Omega_i(t), \quad \forall i = 1, 2, 3, \dots, N.$$

Ω_T as an entire computational domain is independent of t . As Ω_f and Ω_i are always depends on time t we denote $\Omega_f(t) = \Omega_f$ and $\Omega_i(t) = \Omega_i$ dropping t in the notations. Where $\partial\Omega_i$ represents the boundary of the i^{th} particle.

Fluid Flow Model

The motion of an incompressible fluid with density ρ_f is governed by the equations of continuity and momentum in the domain Ω_f as,

$$\rho_f \left[\frac{\partial u}{\partial t} + u \cdot \nabla u \right] - \nabla \cdot \sigma = 0, \quad \nabla \cdot u = 0, \quad \forall t \in (0, T), \quad (1)$$

σ is the total stress tensor in the fluid phase defined as

$$\sigma = -pI + \mu_f[\nabla u + (\nabla u)^T], \quad (2)$$

where, u is the fluid velocity, p is the pressure, μ_f is the coefficient of viscosity and I is the identity tensor.

Particle Motion Model

The rigid particles are free and allowed to move in the fluid domain. The particles have both translational and rotational motion under the action of gravity. Each particle experiences two type of forces one is hydrodynamic force due to fluid-particle interaction and other is collision force due to particle-particle and particle-wall interactions. The motion of solid particles is governed by the Newton-Euler equations [12, 15] i.e. if U_i and ω_i are the translational and angular velocities of the i^{th} particle respectively, then they satisfy,

$$M_i \frac{dU_i}{dt} = (\nabla M_i)g + F_i + F_i', I_i \frac{d\omega_i}{dt} + \omega_i \times (I_i \omega_i) = \tau_i, \quad (3)$$

where M_i is the mass of the i^{th} particle and if M_f is the mass of fluid occupying the same volume as M_i then ΔM_i is given by the mass difference between the mass of the i^{th} particle M_i and the mass of the fluid M_f ,

$$\Delta M_i = M_i - M_f,$$

F_i represents resultant hydrodynamic i.e. drag and lift forces acting on the i^{th} particle, F_i' are the collision forces acting on the i^{th} particle due to particle-particle and particle-wall collision, I_i is the moment of inertia tensor of the i^{th} particle, τ_i is the resultant torque acting about the center of mass of the i^{th} particle and g is the gravitational acceleration.

Fluid-Particle Interactions

Hydrodynamic Forces and Torque

The hydrodynamic drag and lift forces F_i and the torque τ_i about the center of mass of the i^{th} particle can be obtained by[16],

$$F_i = (-1) \int_{\partial\Omega_i} (\sigma \cdot n) d\Gamma_i, \tau_i = (-1) \int_{\partial\Omega_i} (X - X_i) \times (\sigma \cdot n) d\Gamma_i, \quad (4)$$

where σ is the total stress tensor in the fluid phase defined by the Eq. (2), X_i is the position of the center of mass of the i^{th} particle, $\partial\Omega_i$ is the boundary of the i^{th} particle and n is the outward drawn unit normal vector on the boundary $\partial\Omega_i$ of the i^{th} particle. Let S be the surface of rigid particles n_s be the inward drawn unit vector normal to the surface S of the particles and \vec{T} is the tangential vector given by,

$$\vec{T} = (n_y, -n_x).$$

Then using Eq. (4) and Eq. (2) the drag and lift forces are calculated as,

$$F_D = \int_S \left(\mu \frac{\partial u_r}{\partial n_s} n_y - p n_x \right) ds, F_L = - \int_S \left(\mu \frac{\partial u_r}{\partial n_s} n_x + p n_y \right) ds, \quad (5)$$

The drag and lift coefficients are given by,

$$C_d = \frac{2F_D}{\rho \bar{U}^2 D}, C_l = \frac{2F_L}{\rho \bar{U}^2 D}, \quad (6)$$

where C_d and C_l are the coefficients of the drag and lift forces respectively, U is the characteristic velocity and D is the characteristic length. From Eq. (5) and Eq. (6), it is clear that surface integral in Eq. (5) can be conducted for the calculation of C_d and C_l .

Momentum Interaction

Let X_i be the position of the center of mass of the i^{th} particle and θ_i be its angle, then the position X_i and angle θ_i can be obtained by integrating the following kinematic equations [12, 14],

$$\frac{dX_i}{dt} = U_i, \frac{d\theta_i}{dt} = \omega_i. \quad (7)$$

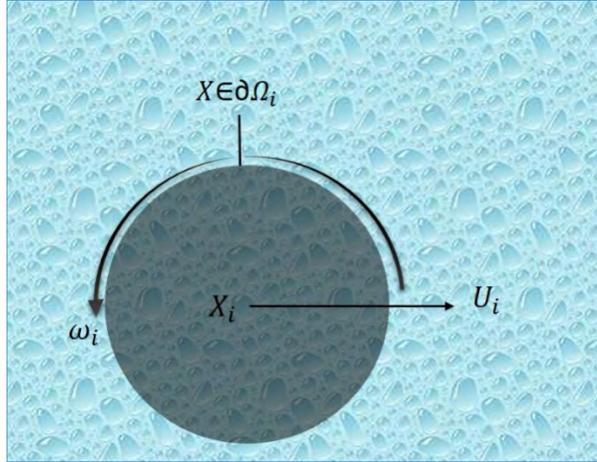


Figure 2. Fluid particle interface.

By the application of no-slip boundary conditions at the interface $\partial\Omega_i$ between the fluid and the i^{th} particle, the velocity $U(X) \forall X \in \Omega_i$ is given by,

$$U(X) = U_i + \omega_i \times (X - X_i), \quad (8)$$

where U_i is the translational velocity of the i^{th} particle and the second term of the sum is the tangential part of the angular velocity of the i^{th} particle

Collision Model

To handle particle-particle and particle-wall interaction, we employed a collision model presented by P. Singh et al. [1]. This is actually a discrete element method (DEM) for modeling the

movement and collision of particles, in which new short range forces of repulsion has been introduced which not only stop the particles from getting very close, but it can also deal with the overlapping of particles when numerical simulations bring these particles very close to each other due to inevitable numerical errors.

Particle-Particle interaction

For the particle-particle collisions, the force of repulsion can be obtained as,

$$F_{i,j}^P = \begin{cases} 0, & \text{for } D_{i,j} > R_i + R_j + \rho, \\ \frac{1}{\epsilon_p} (X_i - X_j)(R_i + R_j + \rho - D_{i,j})^2, & \text{for } R_i + R_j \leq D_{i,j} \leq R_i + R_j + \rho, \\ \frac{1}{\epsilon_p'} (X_i - X_j)(R_i + R_j - D_{i,j}), & \text{for } D_{i,j} \leq R_i + R_j, \end{cases} \quad (9)$$

where X_i and X_j are the coordinates of the centers of the i^{th} and j^{th} particle respectively and $D_{i,j} = |X_i - X_j|$ is the distance between the centers of mass of the particles. R_i and R_j are the radius of the i^{th} and j^{th} particle respectively. ρ is called the range of the force of repulsion and usually $\rho = 0.5 \sim 2.5\Delta h$, where Δh is the size of the mesh element. ϵ_p and ϵ_p' are small positive stiffness parameters for particle-particle collisions. The selection for the values of stiffness parameters ϵ_p and ϵ_p' are such that they do not cause a discontinuity or singularity. For a sufficiently viscous fluid, and $\rho \simeq \Delta h$ as well as ρ_i / ρ_f are of order 1 (ρ_i is the density of the i^{th} particle and ρ_f is the fluid density), then the values of $\epsilon_p \simeq (\Delta h)^2$ and $\epsilon_p' \simeq \Delta h$ are used in the calculations

Particle-Wall interaction

For the particle-wall collisions, the corresponding force of repulsion is expressed by,

$$F_i^W = \begin{cases} 0, & \text{for } D_i' > 2R_i + \rho, \\ \frac{1}{\epsilon_w} (X_i - X_i')(2R_i + \rho - D_i')^2, & \text{for } 2R_i \leq D_i' \leq 2R_i + \rho, \\ \frac{1}{\epsilon_w'} (X_i - X_i')(2R_i - D_i')^2, & \text{for } D_i' \leq 2R_i, \end{cases} \quad (10)$$

where X_i' is the coordinate of the center of mass of the nearest imaginary particle P_i' imagined on the boundary wall Γ with respect to the i^{th} particle, $D_i' = |X_i - X_i'|$ is the distance between the center of the imaginary particle P_i' and the mass center of i^{th} particle. ϵ_w and ϵ_w' are small positive stiffness parameters for particle-wall collisions, usually their values can be taken as $\epsilon_w = \epsilon_p/2$ and $\epsilon_w' = \epsilon_p'/2$ in the calculations.

Numerical Implementation

Multigrid Solver

Multigrid methods were invented for PDEs like Poisson's equation. However, these methods also work on large number of problems. Contrary to other iterative methods, the convergence rate of multigrid method is independent of problem size. For an introduction to multigrid, the reader is

referred to the book of Hackbush[17]. We can use multigrid approach to solve the fluid particle interaction problems which is based on number of grids, obtained by regularly refining the coarse mesh. For CFD problems, multigrid is one of the fastest linear solver [18]. In multigrid, restriction is applied to the residual after smoothing on all mesh levels, and direct sparse linear solver [19] is used to get the solution on coarsest grid, if the number of degrees of freedom is sufficiently small. Then prolongation is applied followed by post-smoothing to obtain better approximation. These steps continue until the iterations of multigrid cycle (V or F-cycle) is finished. With the help of some operators, here we explained how multigrid works on a problem using different grid levels.

The multigrid procedure to solve the linear system $A_i u_i = b_i$, is presented in the following steps.

- Multigrid starts with an initial guess on fine grid level i , i.e., u_i^0 and executes pre-smoothing to obtain more accuracy

$$u_i^{j+1} = S_i(u_i^j), j = 0, \dots, m - 1.$$

where S_i is the smoothing operator and it computes the first improved approximation to the system $A_i u_i = b_i$.

- The high frequency of the residual is sufficiently smoothed by pre-smoothing so that the remaining error shows high frequency on a coarser grid,

$$r_{i-1} = R_i^{i-1}(b_i - A_i u_i^m),$$

where R_i^{i-1} is the restriction operator from a finer grid to coarser grid which gives coarser grid approximation.

- Solve the system

$$A_{i-1} u_{i-1}^* = r_{i-1},$$

on the coarser grid to get the correction u_{i-1}^* .

- Prolongate the correction u_{i-1}^* to the finer grid level and apply

$$u_i^{m+1} = u_i^m + \alpha_i P_{i-1}^i u_{i-1}^*,$$

where P_{i-1}^i is the prolongation operator from coarser grid to next finer grid and α_i is the damping parameter.

- Execute the post smoothing steps to get the final solution u_i^{m+1+n} .

By applying these steps recursively on different grid levels, faster reduction of error is achieved. Appropriate algorithms for restriction, prolongation, smoother and solver components have to be chosen to get accuracy and efficiency. Fig. 3 shows a schematic diagram of multigrid V-cycle (MGV), multigrid W-cycle (MGW) and multigrid F-cycle (MGF).

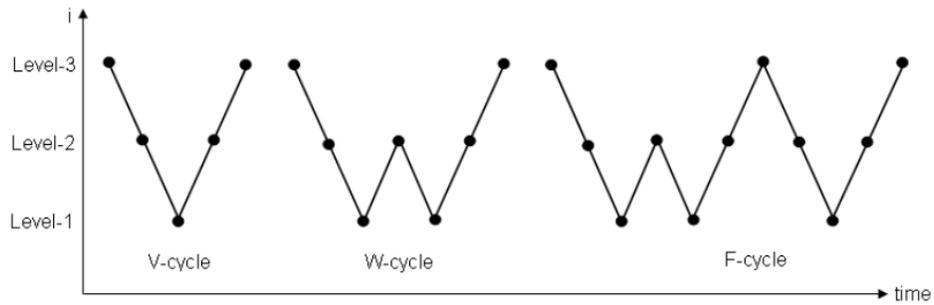


Figure 3. MGV, MGW and MGF cycles

The basic idea for the construction of multigrid V-cycle (MGV) or multigrid F-cycle (MGF) algorithm is as follows.

Algorithm 1. Multigrid V-Cycle

```

function  $MGV(b^i, u^i)$ 
  replace the approximate solution  $u^i$  of the system  $A_i u_i = b_i$  with an improved solution
  if  $\bar{r} = 1$ 
    compute the exact solution  $u^1$ 
    return  $u^1$ 
  else
    •  $u^i = S(b^i, u^i)$ ,
    •  $r^i = A^i \cdot u^i - b^i$ ,
    •  $d^i = P(MGV(R(r^i), 0))$ ,
    •  $u^i = x^i - d^i$ ,
    •  $u^i = S(b^i, u^i)$ ,
    return  $u^i$ 
  end if

```

Algorithm 2. Full Multigrid (FMG)

```

function  $MGF(b^i, u^i)$ 
  return an accurate solution  $u^i$  of the system  $A_i u_i = b_i$ 
  solve  $A_1 u_1 = b_1$  exactly to get  $u^1$ 
  for  $i = 2$  to  $k$ 
     $u^i = MGV(b^i, P(u^i))$ 
  end for

```

Fictitious Boundary Method

Several approaches have been presented using fictitious boundary method to deal with the particles in the fluid and to calculate the hydrodynamic forces acting on the particles. Glowinski, Joseph and coauthors [8] described a semi-implicit approach to calculate the drag and the lift forces acting on the particles in the fluid and study their movement in the fluid. Patankar, Singh, Joseph, Glowinski and co-authors [20] also used an implicit scheme for the particle treatment. Wan and Turek[13] proposed an explicit way to treat the moving particles inside the fluid and the explicit calculation of the drag and the lift forces acting on the particles. Here we are presenting a brief description of multigrid FEM fictitious boundary method, for details reader is referred to the articles[11-13].

This multigrid fictitious boundary method is based on FEM background grid which covers the whole computational domain Ω_T . The grid can be chosen independent of solid particles arbitrary shape, size and number. The multigrid FBM starts with a coarse mesh which may already describes the geometrical details of particles $\Omega_i (i = 1, 2, \dots, N)$ and a boundary parametrization with respect to the boundary conditions of Eq. (8) which sufficiently depict all fine-scale structures of the particles. The internal solid objects are introduced in the corresponding components in all matrices and vectors in the solution process as unknown degrees of freedom. Then, the extra conditions arising due to the interior objects are incorporated implicitly in all the iterative solution steps. Hence, by using Eq. (8), the computations can be carried out on the whole domain Ω_T for the fluid. The FBM has a considerable advantage that the computational domain does not require to be changed with time, and no re-meshing is required. More precisely, the mesh and the flow features can be handled independent of each other [11, 13]. Hence using the FBM, the domain of definition of the fluid velocity u is extended according to Eq. (8) which can be seen as an additional constraint to the Navier-Stokes equations.

$$\begin{cases} \nabla \cdot u = 0, & \forall X \in \Omega_T, \\ \rho_f \left(\frac{\partial u}{\partial t} + u \cdot \nabla u \right) - \nabla \cdot \sigma = 0, & \forall X \in \Omega_f, \\ u(X) = U_i + \omega_i \times (X - X_i), & \forall X \in \bar{\Omega}_i, \quad i = 1, 2, 3, \dots, N. \end{cases} \quad (11)$$

Calculation of Hydrodynamic Forces and Torque

In particulate flows, the calculation of the hydrodynamic forces and torque acting on the particles moving in the fluid is very important for the study of interactions between the fluid and the particles. Let $\Omega_T = \Omega_f \cup \{\Omega_i\}_{i=1}^N$ be the total computational domain including domain occupied by the particles and the fluid. Let n be the unit vector drawn normal to the boundary $\partial\Omega_i$ of i^{th} particle pointed outward to the flow region. To calculate the hydrodynamic forces F_i acting on the surface of the i^{th} particle and the torque τ_i acting about the mass center of the i^{th} particle, the surface integrals on the wall surface of the particle given by Eq. (4) should be conducted. Wan and Turek[13] proposed a volume integral approach rather than the surface integral approach for the calculation of hydrodynamic forces and torque acting on the solid bodies moving in the fluid. In [13], they replaced the surface integral in Eq. (4) by a volume integral, which is computationally less expensive, by defining an auxiliary function α_i ,

$$\alpha_i = \begin{cases} 1, & X \in \Omega_i, \\ 0, & X \in \Omega_f, \end{cases} \quad (12)$$

The gradient of α_i , is zero everywhere except on the wall surface of the i^{th} particle, and at the wall surface the gradient equals to the unit vector n normal to the wall surface of the i^{th} particle, i.e,

$$n = \nabla\alpha_i. \quad (13)$$

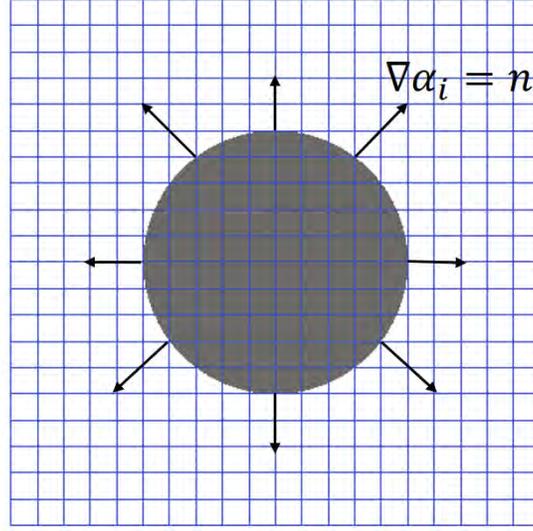


Figure 4. Cells where $\alpha_i = 1$ and $\nabla\alpha_i = n$ on the particle's boundary.

Then for the calculation of hydrodynamic forces and torque, the surface integral in Eq. (4) can be replaced by volume integral as,

$$F_i = (-1) \int_{\Omega_T} (\sigma \cdot \nabla\alpha_i) d\Omega, T_i = (-1) \int_{\Omega_T} (X - X_i) \times (\sigma \cdot \nabla\alpha_i) d\Omega. \quad (14)$$

The volume integral over each element covers the total domain Ω_T can be truly calculated with the help of a standard Gaussian quadrature of sufficiently higher order. As the gradient of α_i , i.e. $\nabla\alpha_i$ is non-zero on the wall surface of the i^{th} particle, therefore the volume integrals in Eq. (14) required to be evaluated only in one layer of mesh cells around the i^{th} particle, which is very effective treatment.

Calculation of Drag and lift coefficients

By using Eq. (2) and Eq. (13), the surface integral in Eq. (5) is replaced by volume integral for the calculation of drag and lift forces acting on the particles in fluid. Such that,

$$F_D = - \int_{\partial\Omega_T} \left[\mu \left(\frac{\partial u}{\partial x} \frac{\partial \alpha}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial \alpha}{\partial y} \right) - p \frac{\partial \alpha}{\partial x} \right] d\Omega, \quad (15)$$

$$F_L = - \int_{\partial\Omega_T} \left[\mu \left(\frac{\partial v}{\partial x} \frac{\partial \alpha}{\partial x} + \frac{\partial v}{\partial y} \frac{\partial \alpha}{\partial y} \right) - p \frac{\partial \alpha}{\partial y} \right] d\Omega, \quad (16)$$

Therefore, through Eq. (15), Eq. (16) and Eq. (6) the drag and lift coefficients C_d and C_l can be calculated by conducting the volume integral over the total domain Ω_T rather than the surface integral conducted over the wall surface of the rigid bodies in Eq. (5).

Simulation Results and Discussions

Flow around a cylinder (benchmark)

For this benchmark case, we test the fluid flow around a fixed circular cylinder. The width and height of the channel is 2.2 and 0.41 respectively. The radius of the circular cylinder is 0.05 and its center is fixed at point (0.2, 0.2). No-slip boundary conditions are imposed at upper and lower walls as well as on the surface of cylinder. The left boundary of the channel is an inlet, at this boundary a parabolic profile is defined for the inflow with maximum velocity $U = 0.3$ at the center of the channel,

$$U(0, y) = \left(\frac{4U_y(0.41 - y)}{(0.41)^2}, 0 \right),$$

where y is along the height of the channel. For outflow, do-nothing boundary condition is defined on the right edge. The fluid density $\rho_f = 1$, where Reynolds number $Re = 20$ with mean velocity $U = 2U(0, y)/3$. We use a locally refined fixed Cartesian mesh as shown in Fig. 5 with number of elements $NEL = 132$ at mesh refinement level $LVL - 1$ (*numberofrefinements*). This coarse mesh is refined by connecting midpoints of opposite edges of an element, so an element divides in to four elements after every refinement. We noticed that at $Re = 20$ the flow turns into stationary. Fig. 6 shows the stationary profiles of pressure, velocity and stream function.

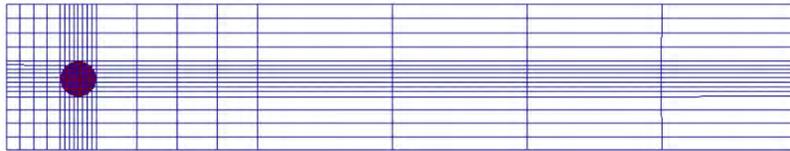


Figure 5. Fixed Cartesian mesh

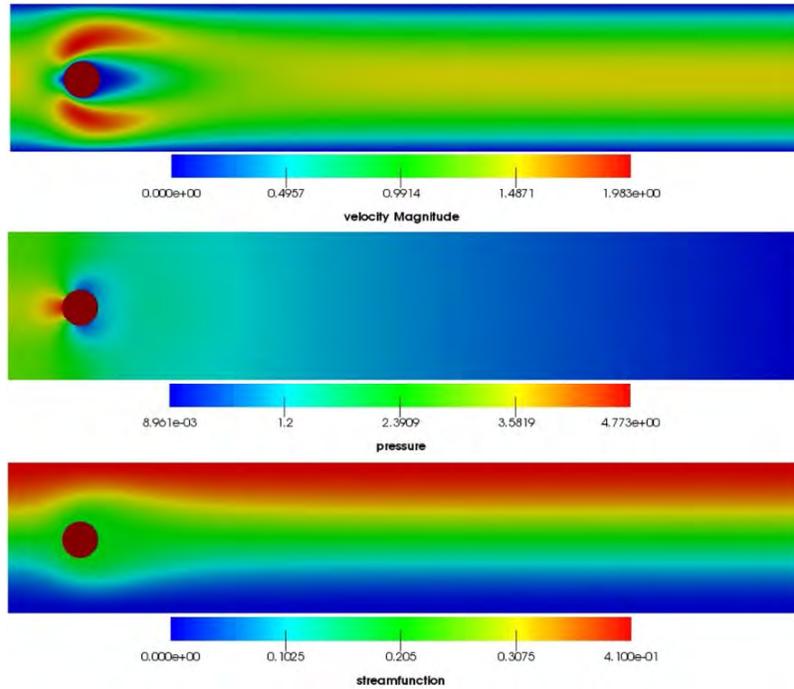


Figure 6. Stationary flow profiles

Table 1. Drag C_d and Lift C_l coefficients

<i>Level</i>	<i>NEL</i>	<i>VEF (%)</i>	C_d	C_l
3	2112	95.493	5.3006	0.0079
4	8448	97.482	5.4211	0.0071
5	33,792	99.472	5.4891	0.0100
6	135,168	99.721	5.5456	0.0098
7	540,672	99.814	5.5573	0.0101
Reference values			5.5795	0.0106

We calculate the drag coefficient C_d and lift coefficient C_l at different mesh refinement levels and the results are presented in table-1. VEF is the ratio of area of cylinder covered by mesh to the actual area of cylinder. All the results are convergent and results of $LVL - 7$ shows a good agreement with the reference values [21]. From the data, it is clear that by increasing mesh refinement level one can achieve the desired accuracy.

A circular particle settling

We now consider a single circular particle of radius $R = 0.125$ in a rectangular channel of width 2 and height 6. At time $t = 0$, the particle is placed at point (1, 4), initially both the particle and incompressible fluid are at rest. As the time passes this particle starts falling downward under the action of gravity with gravitational acceleration $g = 980$. The density of the fluid $\rho_f = 1$ and its viscosity $\nu = 0.1$ or 0.01 , where the density of particle ρ_i is selected as 1.25 or 1.5. We use a fixed

equidistant mesh with number of elements $NEL = 432$ at mesh refinement level $LVL = 1$. The simulations are done on refinement level $LVL = 5$ and $LVL = 6$ having $NEL = 110, 592$ and $NEL = 442, 368$ respectively.

The maximum Reynolds number for a particle during its settling in the fluid is presented in *Table – 2*. The maximum value of the Reynolds number is obtained from the expression $Re = \sqrt{(u(t)^2 + v(t)^2)}.D.\rho_i/\nu$, where $u(t)$ and $v(t)$ are the x and y – *component* of particle velocity at time t respectively and D is the diameter of the particle. We observed that, when the particle is dense and fluid viscosity is small the value of maximum Reynolds number is high, the particle moves much faster and the symmetry breaking is more obvious. For higher fluid viscosity, the flow stays laminar whereas for smaller viscosity flow become turbulent and unstable. The present results show a good agreement with the reference results presented in Ref. [22].

Table 2. Maximum Reynolds numbers

<i>Level</i>	$\nu = 0.1$		$\nu = 0.01$	
	$\rho_i = 1.25$	$\rho_i = 1.5$	$\rho_i = 1.25$	$\rho_i = 1.5$
5	16.679	31.535	273.58	464.98
6	16.307	30:686	280:36	482.28

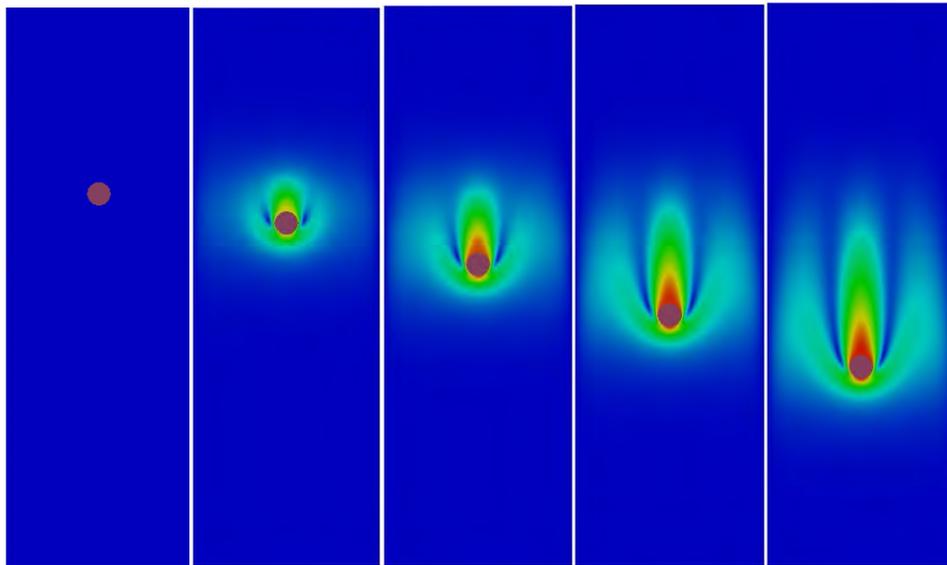


Figure 7. Snapshots of single particle settling at time $t = 0, 0.1, 0.2, 0.3$ and 0.4 respectively.

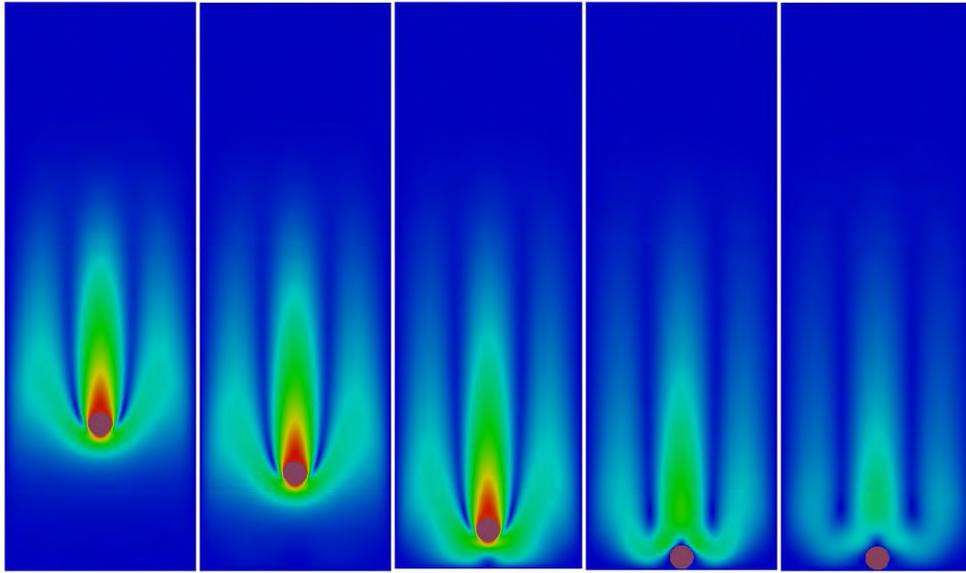


Figure 8. Snapshots of single particle settling at time $t = 0.5, 0.6, 0.7, 0.8$ and 0.9 respectively.

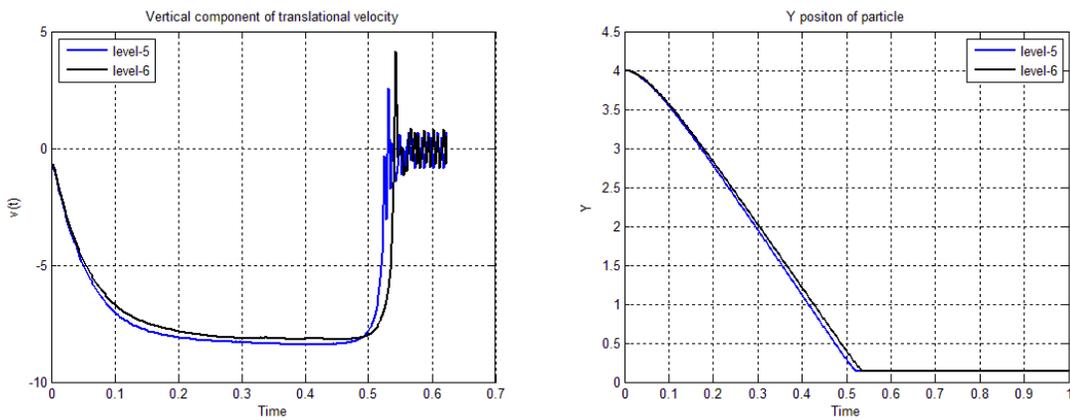


Figure 9. A circular particle settling in fluid with $\nu = 0.1$, $\rho_i = 1.5$, time histories of vertical component of velocity (left) and vertical component of its center of mass (right).

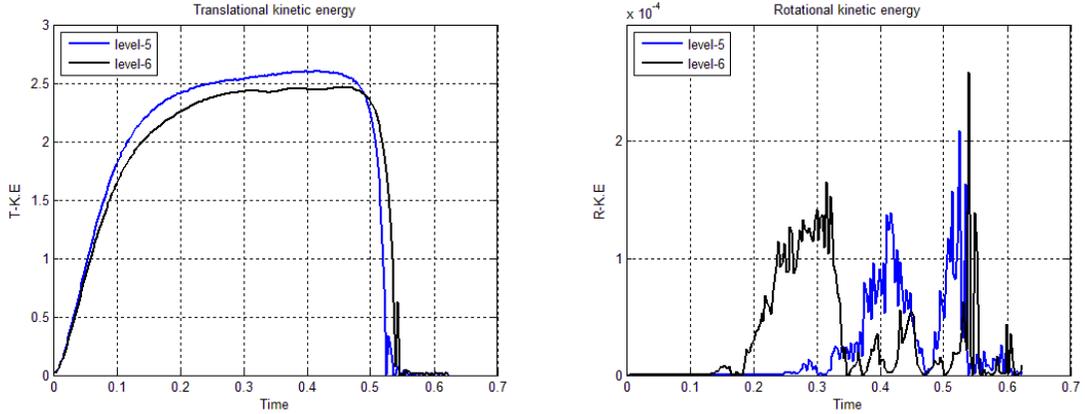


Figure 10. A circular particle settling in fluid with $\nu=0.1$, $\rho_i = 1.5$, time histories of its translational (left) and rotational kinetic energy (right).

Fig. 7 and Fig. 8 shows the snapshots of velocity fields during the particle sedimentation at different time steps. The plots of vertical component of particle velocity and position are shown in Fig. 9 when the particle density $\rho_i = 1.5$ and fluid viscosity $\nu = 0.1$. Whereas the Fig. 10 shows the translational kinetic energy and rotational kinetic energy of particle with respect to time.

A big particle diving in to small particles

The intent of this case is to show that this method is suitable for more complex problems and it can handle large number of particles very well. We consider 2000 small circular particles each of radius 0.025 and a big particle of radius 0.2 in a rectangular cavity of width 4 and height 2. The small particles are settled at the bottom of the cavity, filled with an incompressible fluid of density $\rho_f = 1$, and the center of the big particle is located at point (0.2, 3.8), as shown in Fig.11. Initially all the particles, big particle and fluid are at rest when time $t = 0$. The range of repulsive force $\rho = 0.01$, where the stiffness parameter $\epsilon_p = 10^{-7}$. A uniform fixed mesh is used with mesh element size $\Delta h = 0.1$ and number of elements $NEL = 800$ at mesh refinement level $LVL = 1$. We simulate three cases with Reynold's number $Re = 100$ by choosing different densities of big particle, where density of small particles is same in each case i.e. $\rho_i = 1.5$. Simulations are carried out on refinement level $LVL = 5$ with $NEL = 204, 800$.

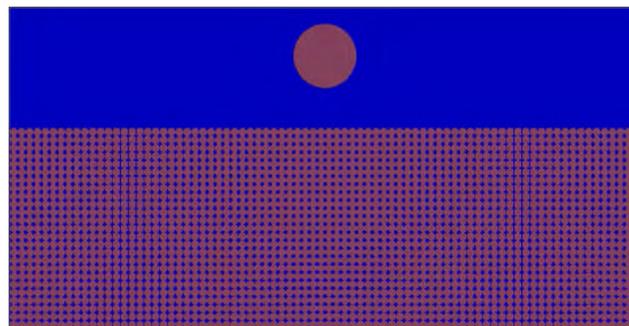


Figure 11. Initial position of particles

Case-1

The density of big particle is $\rho_s = 1.1$, less than the density of small particles ρ_i . From Fig.12, we can see that the big particle dig into small particles and then emerge out quickly and stay on top of small particles because of the strong blockage effect of the small particles

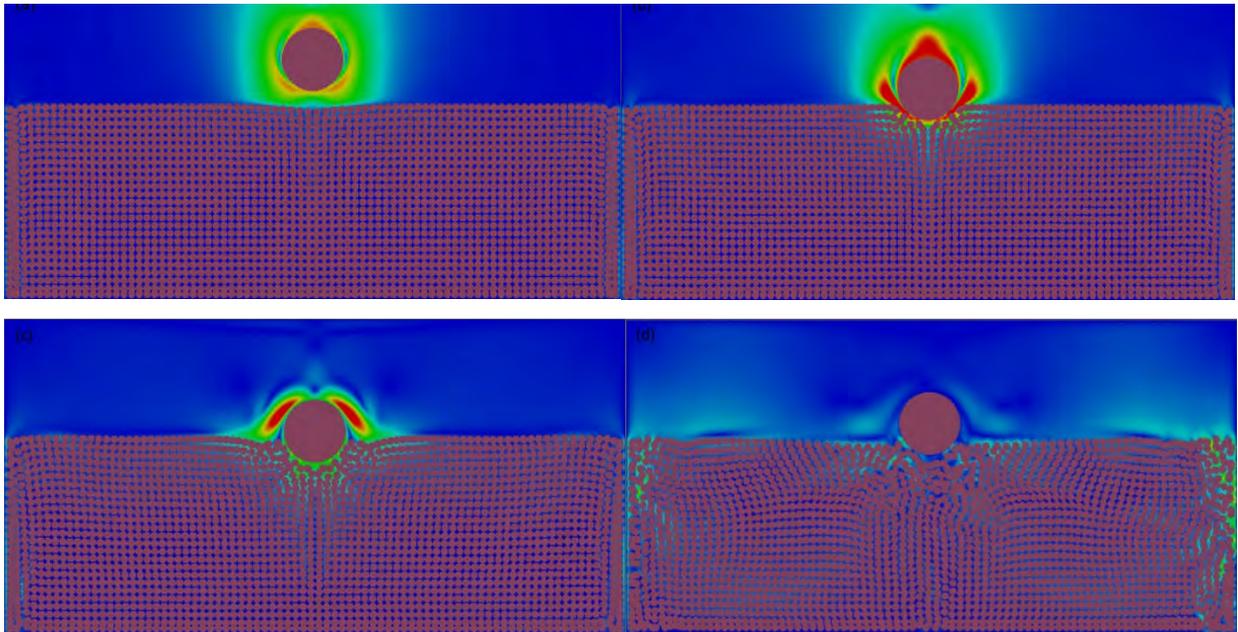
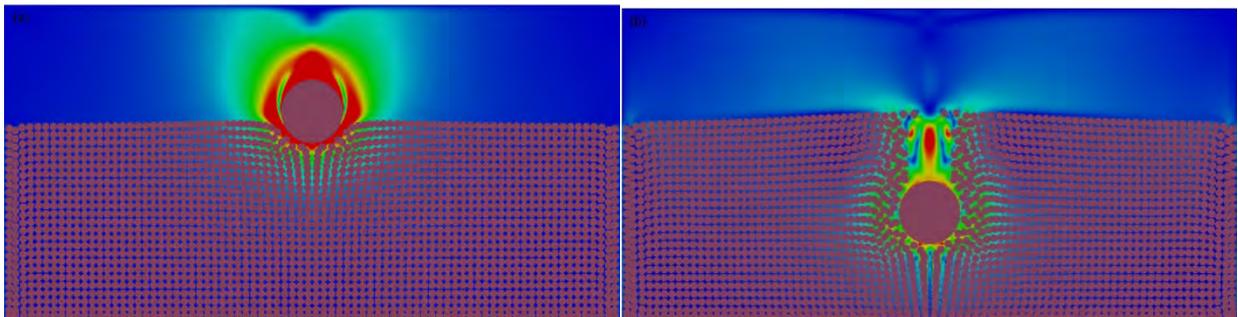


Figure 12. Velocity fields at different time instants.

Case-2

The density of big particle is $\rho_s = 1.5$, equal to the density of small particles ρ_i . The big particle dives into small particles and penetrate slowly into them reaches the bottom of the cavity and at the end settles there, see Fig. 13.



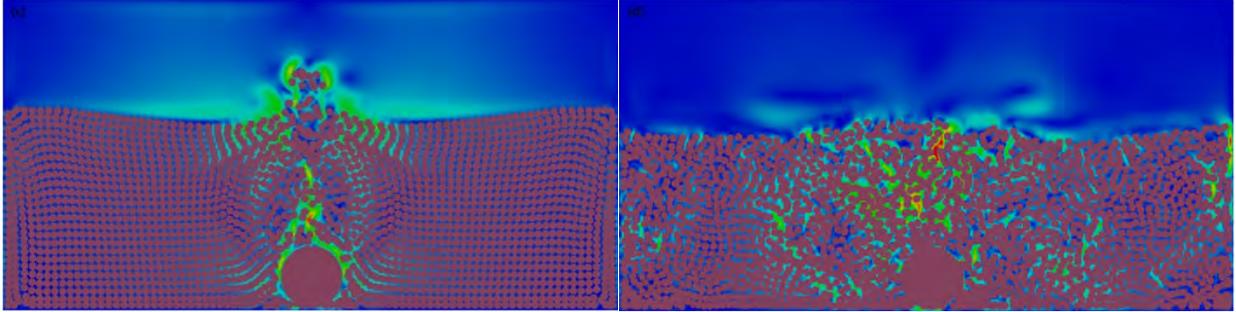


Figure 13. Velocity fields at different time instants.

Case-3

The density of big particle is $\rho_s = 2$, greater than the density of small particles ρ_i .

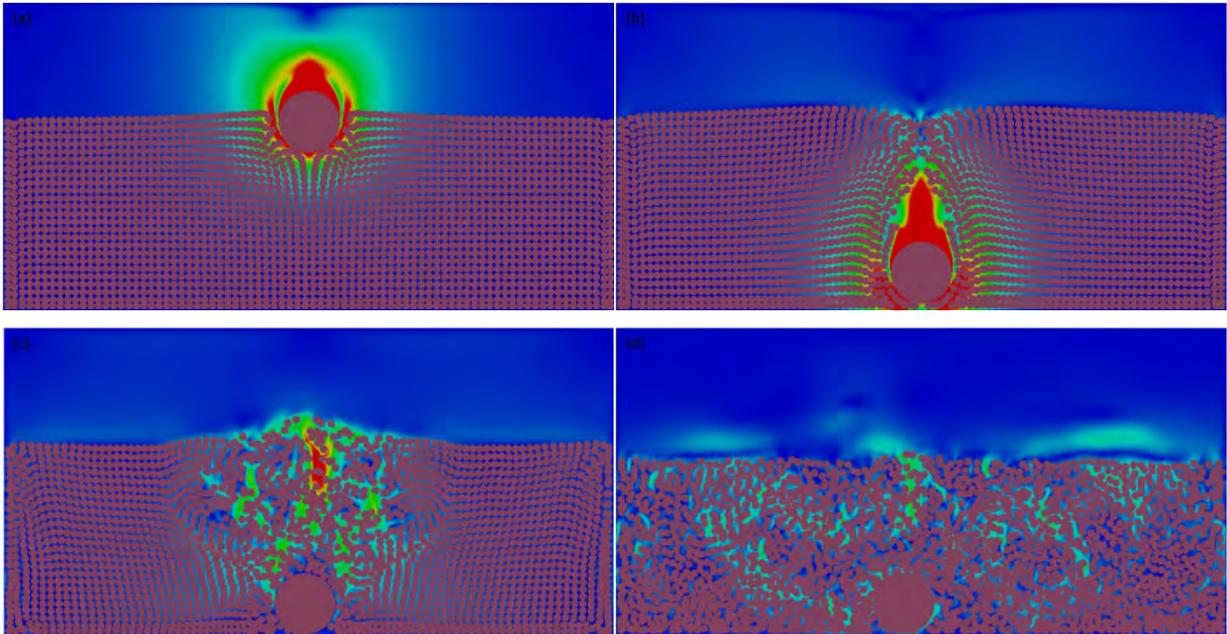


Figure 14. Velocity fields at different time instants.

Fig. 14 shows the big particle push the small particles away from itself and continuously fall until it hits the bottom. This big particle generates some irregular waves at the interface between small particles and the fluid.

Sedimentation of 10,000 particles

Finally, we consider 10,000 circular particles in a rectangular cavity of width 5 and height 8. The particles are placed in rows at the top of the cavity and in each row there are 100 particles. The radius of each particle is 0.025 and initially at time $t = 0$ the fluid and all the particles are at rest. The density of incompressible fluid $\rho_f = 1$, density of solid particles $\rho_i = 1.5$ and Reynolds $Re = 100$. The range of repulsive force $\rho = 0.01$ where the stiffness parameter $\epsilon_p = 10^{-5}$. A uniform fixed mesh is used with mesh element size $\Delta h = 0.00625$ and number of elements $NEL = 1,024,000$ at mesh refinement level $LVL = 5$.

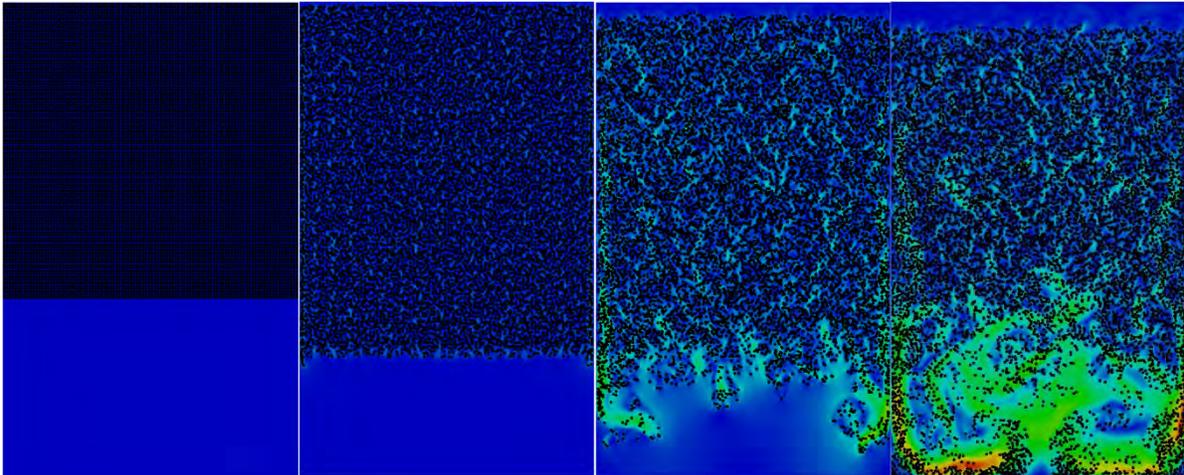


Figure 15. Snapshots of sedimentation of 10,000 particles at time $t = 0$, $t = 1$, $t = 2$, and $t = 3$ respectively.

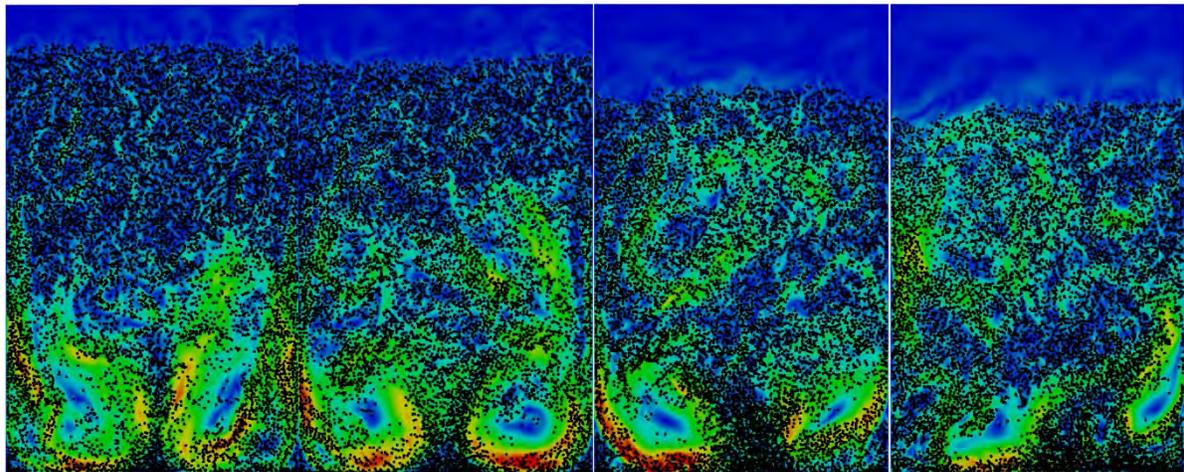


Figure 16. Snapshots of sedimentation of 10,000 particles at time $t = 4$, $t = 5$, $t = 6$, and $t = 7$ respectively.

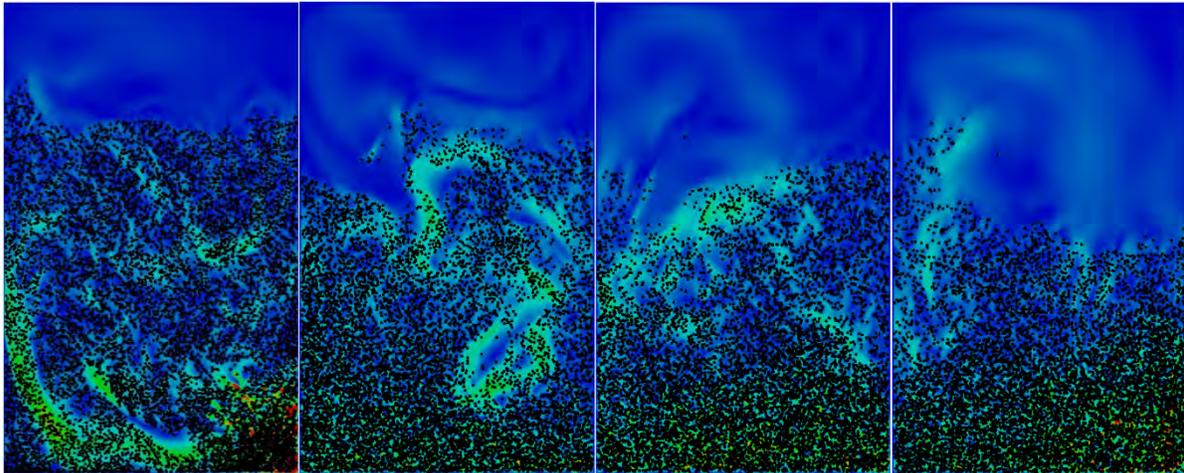


Figure 17. Snapshots of sedimentation of 10,000 particles at time $t = 8$, $t = 9$, $t = 10$, and $t = 11$ respectively.

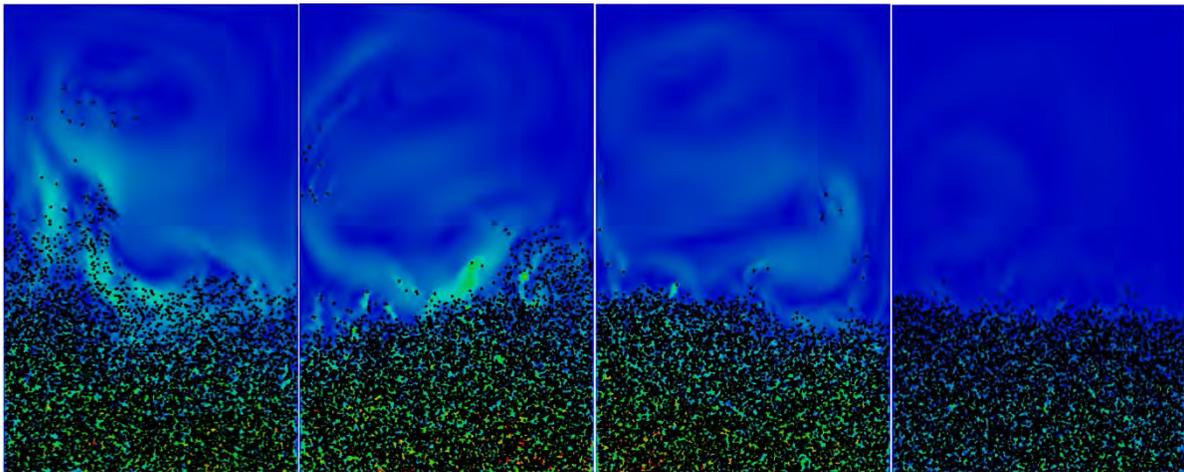


Figure 18. Snapshots of sedimentation of 10,000 particles at time $t = 12$, $t = 13$, $t = 14$, and $t = 15$ respectively.

Fig.15 to 18 shows the snapshots of the velocity fields for 10,000 settling particles at different time steps. The development of Rayleigh-Taylor instability is clearly shown in these pictures. This instability develops into a fingering and text-book phenomenon, and many symmetry breaking and other bifurcation phenomena including drafting, kissing and tumbling take place at various scales in space and time. We can see that many complex vortices of different size have been formed, these vortices pull the particles down and mix them with each other. Some very strong eddies are formed and we can see that these eddies boost the particles and push them almost back to the top of the channel. At the end, all the particles settle down at the bottom of channel and the fluid again comes to rest.

Conclusions

In this paper, we have presented a direct numerical simulation technique multigrid FEM fictitious boundary method for the 2D simulation of solid-liquid two phase flows. We successfully

simulate benchmark case of flow around a fixed cylinder and perform numerical experiments to examine single particle settling in a channel. The accuracy of the presented method has been proved by comparing our results with the corresponding reference results available in literature. Further we simulate more complex problems with large number of particles to test the efficiency of our method and inspect its potential to simulate real particulate flows. Results from the numerical examples of one big particle diving into 2000 small particles and sedimentation of 10,000 particles shows that multigrid fictitious boundary method is computationally less expensive and it can handle large number of particles easily. We conclude that the presented method is computationally cheap and its implementation is very simple and straight forward, as it treats the fluid part, the calculation of hydrodynamic forces and the motion of particle in a sub-sequential way. Finally, the presented method is based on Navier-Stokes equations and it is quite simple to extend it to study heat transfer in solid-liquid two phase flows using Boussinesq approximation and will be discussed in our upcoming article.

References

- [1] P. Singh, T. Hesla, and D. Joseph, "Distributed Lagrange multiplier method for particulate flows with collisions," *International Journal of Multiphase Flow*, vol. 29, no. 3, pp. 495-509, 2003.
- [2] H. H. Hu, D. D. Joseph, and M. J. Crochet, "Direct simulation of fluid particle motions," *Theoretical and Computational Fluid Dynamics*, vol. 3, no. 5, pp. 285-306, 1992.
- [3] H. H. Hu, "Direct simulation of flows of solid-liquid mixtures," *International Journal of Multiphase Flow*, vol. 22, no. 2, pp. 335-352, 1996.
- [4] B. Maury, "Direct simulations of 2D fluid-particle flows in biperiodic domains," *Journal of computational physics*, vol. 156, no. 2, pp. 325-351, 1999.
- [5] B. Maury, "Characteristics ALE method for the unsteady 3D Navier-Stokes equations with a free surface," *International Journal of Computational Fluid Dynamics*, vol. 6, no. 3, pp. 175-188, 1996.
- [6] J. Feng, H. H. Hu, and D. D. Joseph, "Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid Part 1. Sedimentation," *Journal of Fluid Mechanics*, vol. 261, pp. 95-134, 1994.
- [7] C. S. Peskin, "Numerical analysis of blood flow in the heart," *Journal of computational physics*, vol. 25, no. 3, pp. 220-252, 1977.
- [8] R. Glowinski, T. Pan, T. Hesla, D. Joseph, and J. Periaux, "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*, vol. 169, no. 2, pp. 363-426, 2001.
- [9] R. Glowinski, "Finite element methods for incompressible viscous flow," *Handbook of numerical analysis*, vol. 9, pp. 3-1176, 2003.
- [10] R. Glowinski, T.-W. Pan, T. I. Hesla, and D. D. Joseph, "A distributed Lagrange multiplier/fictitious domain method for particulate flows," *International Journal of Multiphase Flow*, vol. 25, no. 5, pp. 755-794, 1999.
- [11] S. Turek, D. Wan, and L. S. Rivkind, "The Fictitious Boundary Method for the implicit treatment of Dirichlet boundary conditions with applications to incompressible flow simulations," in *Challenges in Scientific Computing-CISC 2002*: Springer, 2003, pp. 37-68.
- [12] D. Wan and S. Turek, "Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method," *International Journal for Numerical Methods in Fluids*, vol. 51, no. 5, pp. 531-566, 2006.
- [13] D. Wan, S. Turek, and L. S. Rivkind, "An efficient multigrid FEM solution technique for incompressible flow with moving rigid bodies," in *Numerical Mathematics and Advanced Applications*: Springer, 2004, pp. 844-853.
- [14] D. Wan and S. Turek, "An efficient multigrid-FEM method for the simulation of solid-liquid two phase flows," *Journal of Computational and Applied Mathematics*, vol. 203, no. 2, pp. 561-580, 2007.
- [15] T. I. Zohdi, *An introduction to modeling and simulation of particulate flows*. SIAM, 2007.
- [16] S. Kim and S. J. Karrila, *Microhydrodynamics: principles and selected applications*. Courier Corporation, 2013.
- [17] W. Hackbusch, *Multi-grid methods and applications*. Springer Science & Business Media, 2013.

- [18] P. Wesseling and C. W. Oosterlee, "Geometric multigrid with applications to computational fluid dynamics," *Journal of Computational and Applied Mathematics*, vol. 128, no. 1, pp. 311-334, 2001.
- [19] T. A. Davis and I. S. Duff, "A combined unifrontal/multifrontal method for unsymmetric sparse matrices," *ACM Transactions on Mathematical Software (TOMS)*, vol. 25, no. 1, pp. 1-20, 1999.
- [20] N. A. Patankar, P. Singh, D. D. Joseph, R. Glowinski, and T.-W. Pan, "A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows," *International Journal of Multiphase Flow*, vol. 26, no. 9, pp. 1509-1524, 2000.
- [21] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher, *Benchmark computations of laminar flow around a cylinder*. Springer, 1996.
- [22] R. Glowinski, "Numerical Methods for Fluids (Part 3). Finite Element Methods for Incompressible Viscous Flow. Vol. 9 of Handbook of Numerical Analysis, PG Ciarlet and JL Lions, eds," ed: North-Holland, Elsevier, Amsterdam etc, 2003.



Biography

M. B. Liu received his B.E. and M. E. degrees from Xi'an JiaoTong University (XJTU), China in 1993 and 1996, respectively, and received his PhD from the National University of Singapore (NUS) in 2003. He is currently a professor at the College of Engineering and the Institute of Ocean Research of Peking University. He authored more than 100 reviewed technical papers and one monograph, with more than 3000 SCI citations. He has received a number of prestigious awards from universities and scientific organizations worldwide, including the 100 Talent Program Award from the Chinese Academy of Sciences (CAS) in 2009, the Young Investigator Award from the Asia Pacific Association of Computational Mechanics (APACM) in 2007 and the Lee Kuan Yew Fellow Award from the Nanyang Technological University (NTU) in 2005. He is interested in computer modeling of complex flows with different coupling effects, and has developed some original numerical methods, both grid-based and meshfree particle-based, for modeling fluid dynamics and fluid-solid interactions, in ocean engineering and hydrodynamics (e.g., wave breaking, sloshing and slamming, water entry and exit), safety engineering (e.g., explosion, underwater explosion, high velocity impact, penetration and structure protection) and advanced manufacturing (e.g., additive manufacturing with powder spread and transport, melting and solidification of metal powder).



Khuram Walayat received his B.Sc. (major in Mathematics and physics) (2006) and M.Sc. Mathematics (2008) degrees from University of the Punjab Lahore, Pakistan. He gets his second master's degree in Mathematical Modeling and Scientific Computing (2014) from Air University Islamabad, Pakistan. Currently he is working with Professor M. B. Liu as a PhD candidate at College of Engineering, Peking University. His research interests are Particulate flows, Fluid solid interaction and heat transfer in two-phase flows.



Z.K. Wang got his bachelor degree from Nanjing University of Aeronautics and Astronautics (NUAA) and received the Second Prize of the National ZhouPeiyuan Mechanics Competition in 2015. He is now a PhD Candidate in Peking University and is developing advanced computational techniques for additive manufacturing under the supervision of Professor M. B. Liu. His research interests include Computational Mechanics and Fluid-Solid Coupling.