Journal of Computational Physics 228 (2009) 2712–2738

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A numerical method for fully resolved simulation (FRS) of rigid particle–flow interactions in complex flows

Sourabh V. Apte [a,*], Mathieu Martin [a], Neelesh A. Patankar [b]

[a] School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR 97331, USA
[b] Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA

ARTICLE INFO

Article history:
Received 20 April 2008
Received in revised form 9 September 2008
Accepted 26 November 2008
Available online 24 December 2008

Keywords:
DNS
Particle-turbulence
Interactions
Fully resolved simulation (FRS)
Fictitious-domain

ABSTRACT

A fictitious-domain based formulation for fully resolved simulations of arbitrary shaped, freely moving rigid particles in unsteady flows is presented. The entire fluid–particle domain is assumed to be an incompressible, but variable density, fluid. The numerical method is based on a finite-volume approach on a co-located, Cartesian grid together with a fractional step method for variable density, low-Mach number flows. The flow inside the fluid region is constrained to be divergence-free for an incompressible fluid, whereas the flow inside the particle domain is constrained to undergo rigid body motion. In this approach, the rigid body motion constraint is imposed by avoiding the explicit calculation of distributed Lagrange multipliers and is based upon the formulation developed by Patankar [N. Patankar, A formulation for fast computations of rigid particulate flows, Center for Turbulence Research Annual Research Briefs 2001 (2001) 185–196]. The rigidity constraint is imposed and the rigid body motion (translation and rotational velocity fields) is obtained directly in the context of a two-stage fractional step scheme. The numerical approach is applied to both imposed particle motion and fluid–particle interaction problems involving freely moving particles. Grid and time-step convergence studies are performed to evaluate the accuracy of the approach. Finally, simulation of rigid particles in a decaying isotropic turbulent flow is performed to study the feasibility of simulations of particle-laden turbulent flows.

Published by Elsevier Inc.

1. Introduction

Many problems in nature and engineering involve two-phase flows where solid particles of arbitrary shape and sizes are dispersed in an ambient fluid (gas or liquid) undergoing time dependent and often turbulent motion. Examples include sediment transport in rivers, fluidized beds, coal-based oxy-fuel combustion chambers, biomass gasifiers, among others. Fully resolved simulation (FRS), wherein all scales associated with the fluid flow and the motion of all particles are directly computed, are of importance to understand the fluid–particle interactions. In these simulations, the hydrodynamic forces between the particles and fluids are obtained from direct solution of the governing equations and are not modeled by any drag or lift coefficients. Therefore, such simulations can be used to develop new and improved drag and lift laws. Specifically, for particle-laden turbulent flows, fundamentally understanding the turbulence modulation due to particles and dispersion of particles due to fluctuations in the fluid flow is important to develop reduced-ordered models for fluid–particle systems.

Considerable work has been done on fully resolved simulations of particles in laminar flows. Hu et al. [2] developed an Arbitrary Lagrangian–Eulerian (ALE)-based finite-element approach on unstructured grids to simulate rigid particles in

[*] Corresponding author. Tel.: +1 541 737 7335; fax: +1 541 737 2600.
E-mail address: sva@engr.orst.edu (S.V. Apte).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2008.11.034

Newtonian and viscoelastic fluids. In this approach, the unstructured grids conform to the immersed rigid objects that move as the particles undergo rigid motion. A new mesh and resulting connectivity is generated when the grid becomes too distorted and an interpolation scheme is used to compute the flow field onto the new mesh. Moving mesh algorithm based on space–time finite-element approach was also developed by Johnson and Tezduyar [3] to calculate falling particles in a tube. Such approaches, although provide an accurate solution at the fluid–particle interface, suffer from the complexity of the moving mesh and regeneration algorithms. Use of these techniques in three-dimensions significantly increase the computational cost and memory requirements.

Several numerical schemes based on use of fixed grids for simulation of the fluid fluid–particle system have been investigated. For example, distributed Lagrange multiplier/fictitious-domain (DLM) based methods [4] and immersed boundary method (IBM) [5,6] have been developed and shown to be very effective in computing fluid–particle systems and fluid–structure interaction problems. Lattice Boltzmann method (LBM) [7] has been developed and effectively used for simulations of rigid as well as deforming particles. Combination of the DLM, direct forcing based IBM, and Lattice Boltzmann methods (termed as *Proteus* was recently developed [8]. A second-order accurate fixed grid method (PHYSALIS [9]) was developed, which gives good solutions for *spherical* particles by using local spectral representations of the solution near a spherical boundary.

The immersed boundary method has traditionally been used for fluid–structure interaction problems wherein the motion of the immersed object is specified (stationary, forced rigid motion, or elastically deforming objects). The approach has been used for turbulent flow simulations at large Reynolds numbers using direction numerical simulations (DNS) or large-eddy simulations (LES). For specified motion of immersed objects, Taira and Colonius [10] proposed a new implementation of the immersed boundary method to achieve second-order accuracy. They compared IBM with fictitious-domain based methods to point out subtle differences when the immersed objects are constrained to undergo specified motion. Uhlmann [11] used IBM with direct forcing method for freely moving rigid particulate flows. Recently, Kim and Choi [12] developed a new immersed boundary method using the conservative form of Navier–Stokes and continuity equations in the non-inertial frame of reference and applied to fluid–structure interactions problems wherein the motion of the immersed objects for specified (forced) and also cases with freely moving rigid particles.

In the DLM method [4], the entire fluid–particle domain is assumed to be a fluid and the flow in the particle domain is constrained to be a rigid-body motion by using a rigidity constraint. Similar to the immersed boundary method, this approach uses fixed background grids and eliminates the need for remeshing and moving meshes. A Lagrange multiplier field in the particle domain is computed by treating the fluid–particle motion implicitly and solving a combined weak formulation. The constraint of rigid body motion is represented by $\mathbf{u} = \mathbf{U} + \boldsymbol{\omega} \times \mathbf{r}$, where $\mathbf{u}$ is the velocity of the fluid inside the particle domain, $\mathbf{U}$ and $\boldsymbol{\omega}$ are the translational and angular velocities of the particle, $\mathbf{r}$ is the position vector of a point within the particle region with respect to the particle centroid. Patankar et al. [13] developed a new formulation, named as the stress-DLM formulation, wherein the rigid body motion was obtained by constraining a deformation rate tensor within the particle region to be zero. As opposed to the original DLM formulation, this approach eliminated the need for $\mathbf{U}$ and $\boldsymbol{\omega}$ as variables from the coupled system of equations and provided simplified approach for simulation of irregular shaped bodies. Both approaches, however, require use of an iterative fractional step scheme and resulted in increased overhead on the solution procedure in the presence of particles. Patankar [1] developed an adapted version of the stress-based DLM formulation, that eliminated the need for an iterative procedure to solve the rigid body projection step. By developing a two-stage fractional step scheme, fast computation of particle-laden fluid flows was presented [1,14] in finite-element and finite-volume frameworks. Recently, an equivalent formulation based on the original DLM approach [4] of rigidity constraint in a finite-element framework was developed by eliminating the need for an iterative solution procedure [15].

Majority of the above approaches have been applied to simulate rigid particulate flows at low Reynolds number laminar flows. In spite of several different numerical schemes, full three-dimensional direct simulations of two-phase turbulent flows are rare. There have been only few three-dimensional studies on fully resolved rigid particles in turbulent flows [16,17] in canonical flow problems. There appears to be no reported study of fully resolved moving particles in complex geometries. Majority of the works using IBM or fictitious-domain based techniques are based on Cartesian, staggered grids. In the present work, a fictitious-domain based approach for the motion of arbitrary shaped rigid particles is developed in a structured, co-located grid finite-volume formulation. The co-located grid formulation is used owing to its flexibility and potential in extending the numerical approach to fixed, unstructured grids and simulations of turbulent flows in complex configurations [18,19]. The approach is based on an efficient numerical algorithm proposed by Patankar [1] to constrain the flow field inside the particle to a rigid body motion. Sharma and Patankar [14] implemented this approach in staggered Cartesian grid solver indicating first-order temporal accuracy in cases with freely moving rigid particles. In the present work, we extend this approach to a time-staggered, co-located formulation and evaluate the order of accuracy of the resultant scheme for rigid particles under freely moving as well as forced motion conditions. Domain decomposition and Message-Passing-Interface (MPI)-based solver parallelization is performed to facilitate simulation of large number of rigid particles. Details of the numerical scheme are outlined and the method is applied to investigate fluid–particle interactions in laminar and turbulent flows. Both forced motion and freely moving rigid particles are simulated.

The paper is arranged as follows. A mathematical formulation of the basic scheme is first described. Numerical implementation of the scheme in a co-located grid, finite-volume framework is provided next. The numerical scheme is validated for flow over a fixed sphere and flow induced by periodically oscillating cylinder as test cases. Freely falling spherical particle at different Reynolds numbers is simulated and results compared with available experimental data. A detailed analysis of the temporal and spatial discretization errors is performed to evaluate the accuracy of the numerical scheme. Unsteady wake interactions between two particles falling under gravity are also investigated and compared with previous numerical studies. Finally, simulation of 125 spherical particles in an isotropic turbulent flow is performed to show the feasibility of the approach to capture multiscale interactions between the particles and unsteady turbulent flows.

2. Mathematical formulation

Let Γ be the computational domain which includes both the fluid ($\Gamma_F(t)$) and the particle ($\Gamma_P(t)$) domains. Let the fluid boundary not shared with the particle be denoted by $\mathcal{B}$ and have a Dirichlet condition (generalization of boundary conditions is possible). For simplicity, let there be a single particle in the domain and the body force be assumed constant so that there is no net torque acting on the particle. The basis of fictitious-domain based approach [4] is to extend the Navier–Stokes equations for fluid motion over the entire domain Γ inclusive of particle regions. The natural choice is to assume that the particle region is filled with a Newtonian *fluid* of density equal to the particle density (ρ_P) and some fluid viscosity (μ_F). Both the fluid and the particle regions will be assumed as incompressible and thus incompressibility constraint applies over the entire region. In addition, as the particles are assumed as rigid, the motion of the material inside the particle is constrained to be a rigid body motion. Several ways of obtaining the rigidity constraint have been proposed [4,13,1,15]. We follow the formulation developed by Patankar [1] which is briefly described for completeness.

The momentum equation for fluid motion applicable in the entire domain Γ is given by:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = -\nabla p + \nabla\cdot\left(\mu_F\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right) + \rho\mathbf{g} + \mathbf{f}, \tag{1}$$

where ρ is the density field, $\mathbf{u}$ the velocity vector, p the pressure, μ_F the fluid viscosity, $\mathbf{g}$ the gravitational acceleration, and $\mathbf{f}$ is an additional body force that enforces rigid body motion inside the particle region Γ_P. For direct numerical simulation of incompressible fluid with constant viscosity, however, the viscous term can be simplified to $\mu_F\nabla^2\mathbf{u}$ using the incompressibility constraint. The density ρ is given as:

$$\rho = \rho_F(1-\Theta_P) + \rho_P\Theta_P;\, \Theta_P = \begin{cases} 0 & \text{in } \Gamma_F, \\ 1 & \text{in } \Gamma_P, \end{cases} \tag{2}$$

where ρ_F and ρ_P are the fluid and particle densities, respectively, Θ_p is the indicator function that assumes a value of unity inside the particle region and zero outside. In general numerical implementations, the indicator function is smeared over a small region (proportional to the grid spacing) around the boundary giving a smooth variation. As the particle moves, so does the indicator function and thus $\mathrm{D}\Theta_p/\mathrm{D}t = 0$ on the particle boundary, where $\mathrm{D}/\mathrm{D}t()$ represents a material derivative. Here we assume that the solid particles experience no-slip boundary conditions; therefore, the transport of the function Θ_p is directly related to the local fluid velocity. The continuity equation in Γ for this variable density Newtonian fluid is given as:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0. \tag{3}$$

Using the definition of ρ, expanding the above equation and noting than $\mathrm{D}\Theta_p/\mathrm{D}t = 0$ on the particle boundaries gives the incompressibility constraint over the entire domain Γ:

$$\nabla\cdot\mathbf{u} = 0, \tag{4}$$

In order to enforce that the material inside the particle region moves in a rigid fashion, a rigidity constraint is required so that it will lead to a non-zero forcing function $\mathbf{f}$ in the particle region. Different ways have been proposed to obtain $\mathbf{f}$. Inside the particle region, the rigid body motion implies vanishing deformation rate tensor [13]:

$$\left.\begin{aligned} &\tfrac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) = \mathbf{D}[\mathbf{u}] = 0, \\ &\Rightarrow \mathbf{u} = \mathbf{u}^{\mathrm{RBM}} = \mathbf{U} + \Omega\times\mathbf{r} \end{aligned}\right\} \quad \text{in } \Gamma_P, \tag{5}$$

where $\mathbf{U}$ and Ω are the particle translation and angular velocities and $\mathbf{r}$ is the position vector of a point inside the particle region from the particle centroid. The vanishing deformation rate tensor for rigidity constraint automatically ensures the incompressibility constraint inside the particle region. The incompressibility constraint gives rise to the scalar field (the pressure, p) in a fluid. Similarly, the tensor constraint $\mathbf{D}[\mathbf{u}] = 0$ for rigid motion gives rise to a tensor field inside the particle region [13]. Distributed Lagrange multiplier (DLM)-based approaches have been proposed to solve for the rigid body motion and impose the rigidity constraint which requires an iterative solution strategy. Patankar [1] proposed an approach that provides the rigidity constraint explicitly, thus reducing the computational cost significantly. Noting that the tensorial rigidity constraint can be reformulated to give [13]:

$$\nabla\cdot(\mathbf{D}[\mathbf{u}]) = 0 \quad \text{in } \Gamma_P, \tag{6}$$

$$\mathbf{D}[\mathbf{u}]\cdot\mathbf{n} = 0 \quad \text{on particle boundary}. \tag{7}$$

A two-stage fractional-step algorithm can be devised to solve the coupled fluid–particle problem [1]. Knowing the solution at time level t^n the goal is to find $\mathbf{u}$ at time t^{n+1}.

1. In this first step, the rigidity constraint force $\mathbf{f}$ in Eq. 1 is set to zero and the equation together with the incompressibility constraint (Eq. 4) is solved by standard fractional-step schemes over the entire domain. Accordingly, a pressure Poisson equation is derived and used to project the velocity field onto an incompressible solution. The obtained velocity field is denoted as $\mathbf{u}^{n+1}$ inside the fluid domain and $\hat{\mathbf{u}}$ inside the particle region.
2. The velocity field in the particle domain is obtained in a second step by projecting the flow field onto a rigid body motion. Inside the particle region:

$$\rho_P\left(\frac{\mathbf{u}^{n+1}-\hat{\mathbf{u}}}{\Delta t}\right)=\mathbf{f}. \tag{8}$$

   To solve for $\mathbf{u}^{n+1}$ inside the particle region we require $\mathbf{f}$. Obtaining the deformation rate tensor from $\mathbf{u}^{n+1}$ given by the above equation and using the Eqs. (6) and (7) we obtain:

$$\nabla\cdot\left(\mathbf{D}[\mathbf{u}^{n+1}]\right)=\nabla\cdot\left(\mathbf{D}\left[\hat{\mathbf{u}}+\frac{\mathbf{f}\Delta t}{\rho}\right]\right)=0, \tag{9}$$

$$\mathbf{D}[\mathbf{u}^{n+1}]\cdot\mathbf{n}=\mathbf{D}\left[\hat{\mathbf{u}}+\frac{\mathbf{f}\Delta t}{\rho}\right]\cdot\mathbf{n}=0. \tag{10}$$

   The velocity field in the particle domain involves only translation and angular velocities. Thus, $\hat{\mathbf{u}}$ is split into a rigid body motion ($\mathbf{u}^{\mathrm{RBM}}=\mathbf{U}+\Omega\times\mathbf{r}$) and residual non-rigid motion ($\mathbf{u}'$). The translational and rotational components of the rigid body motion are obtained by conserving the linear and angular momenta and are given as:

$$M_P\mathbf{U}=\int_{\Gamma_P}\rho\hat{\mathbf{u}}\,d\mathbf{x}, \tag{11}$$

$$\mathcal{I}_P\Omega=\int_{\Gamma_P}\mathbf{r}\times\rho\hat{\mathbf{u}}\,d\mathbf{x}, \tag{12}$$

   where M_P is the mass of the particle and $\mathcal{I}_P=\int_{\Gamma_P}\rho[(\mathbf{r}\cdot\mathbf{r})\mathbf{I}-\mathbf{r}\otimes\mathbf{r}]\,d\mathbf{x}$ is the moment of inertia tensor. Knowing $\mathbf{U}$ and Ω for each particle, the rigid body motion inside the particle region $\mathbf{u}^{\mathrm{RBM}}$ can be calculated.
3. The rigidity constraint force is then simply obtained as $\mathbf{f}=\rho(\mathbf{u}^{\mathrm{RBM}}-\hat{\mathbf{u}})/\Delta t$. This sets $\mathbf{u}^{n+1}=\mathbf{u}^{\mathrm{RBM}}$ in the particle domain. Note that the rigidity constraint is non-zero only inside the particle domain and zero everywhere else. This constraint is then imposed in a third fractional step and using Eq. (8).

In practice, the fluid flow near the boundary of the particle (over a length scale on the order of the grid size) is altered by the above procedure owing to the smearing of the particle boundary. The key advantage of the above formulation is that the projection step only involves straightforward integrations in the particle domain. A similar approach was recently proposed in a finite-element framework by Veeramani et al. [15].

3. Numerical formulation

In this work, the governing equations (Eqs. (1) and (3)) are reformulated into conservative form to obtain better conservation properties and accuracy for unsteady, turbulent flows. A co-located grid formulation on Cartesian grids is used. The approach can be readily extended to unstructured grids and complex configurations. In the following sections, representation and transformations of rigid body motion, interpolation schemes between the particle material points and the background grid, the discretization of the numerical scheme, and the complete algorithm are described.

3.1. Material volume representation of a particle

We represent a particle by introducing material volumes (MV) or points within the particle domain. These volumes can be thought of as "sub-particles" having a specific shape and density. Here, we assume that the material volumes are cubic elements and have same density as the particle itself. However, arbitrary shapes of the material volumes along with varying densities can be easily assigned to represent a complex shaped particle with non-uniform material properties. For rigid body motion of the particle there is no relative velocity between the material volumes and the particle centroid. The relative position (or connectivity) between the material volume centroids and the particle centroid is not necessary. The use of material volumes over the entire particle domain (as opposed to only around the particle surface) is used as it simplifies volume integrations needed in the present scheme.

Fig. 1 shows examples of the material volumes for a particle with circular cross-section. In Fig. 1(a), uniform material volumes arranged in a structured lattice are created, whereas Fig. 1(b) shows body-fitted unstructured material volumes. In the first approach, the boundary of the rigid body is represented in a stair-stepped fashion, however, it is straight forward to create the material volumes using a bounding-box algorithm:

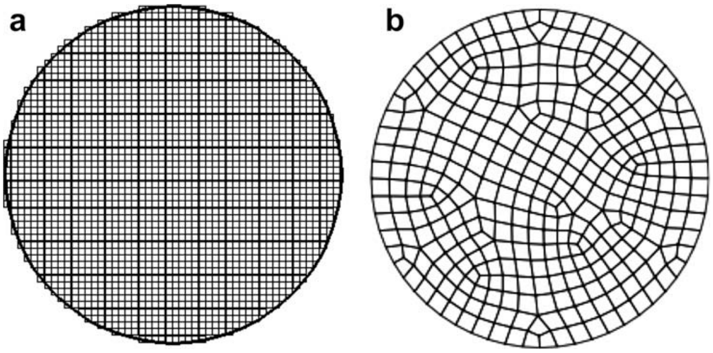


Fig. 1. Schematic of material volume for a circular interface: (a) uniform, stair-stepped grid, (b) body-fitted unstructured grid.

1. Determine the bounding box for the particle based on its surface representation.
2. Generate cubic grid within the bounding box.
3. Use distance searches to determine if the centroid of the control volume lies within the bounding surface of the particle.
4. Eliminate points outside the particle domain.

The total mass of the material volumes generated will be exactly equal to the mass of the particle if the surface of the particle aligns with the grid. The stair-stepped surface representation, however, results in an error in the total mass of the material volumes compared to the original shape. This error reduces with increase in the total number of material volumes per particle. A more complex grid generation process (Fig. 1(b)) and/or Delaunay triangulation is necessary to accurately represent the surface of the particle and one may use standard body-fitted grid generation tools. In the present work, we follow the stair-stepped approach owing to its simplicity.

3.2. Interphase interpolations

Any property defined at the material volumes within the particle can be projected onto the background grid by using interpolation functions. Use of simple linear interpolations may give rise to unphysical values within the particle domain (e.g. volume fractions greater than unity) [14] and may give rise to numerical oscillations in the particle velocity. In order to overcome this, a smooth approximation of the quantity can be constructed from the material volumes using interpolation kernels typically used in particle methods [20]:

$$\Phi^{\Delta}(\mathbf{x}) = \int \Phi(\mathbf{y})\xi^{\Delta}(\mathbf{x}-\mathbf{y})\,d\mathbf{y}, \tag{13}$$

where Δ denotes grid resolution. The interpolation operator can be discretized using the material volume centroids as the quadrature points to give

$$\Phi^{\Delta}(\mathbf{x}) = \sum_{M=1}^{N} V_M \Phi(\mathbf{X}_M)\xi^{\Delta}(\mathbf{x}-\mathbf{X}_M), \tag{14}$$

where $\mathbf{X}_M$ and V_M denote the coordinates and volume of the material volumes, respectively and the summation is over all material volumes for a particle. For example, in order to compute particle volume fraction, $\Phi(\mathbf{X}_M)$ will be unity at all material points. This gives unity volume fraction within the particle domain and zero outside the particle. In order to conserve the total volume of the particle as well as the total force/torque exerted by the particle on the fluid, the interpolation kernel should at least satisfy

$$\sum_{M=1}^{N} V_M \xi^{\Delta}(\mathbf{x}-\mathbf{X}_M) = 1, \tag{15}$$

$$\sum_{M=1}^{N} V_M (\mathbf{x}-\mathbf{X}_M)\xi^{\Delta}(\mathbf{x}-\mathbf{X}_M) = 0. \tag{16}$$

Several kernels with second-order accuracy include Gaussian, quartic splines, etc. A kernel with compact support requiring only the immediate neighbors of a control volume has been designed and used in immersed boundary methods [21]. For uniform meshes with resolution Δ it utilizes only three points in one dimension and gives the sharpest representation of the particle onto the background mesh:

$$\xi^{\Delta}(\mathbf{x}-\mathbf{X}_M) = \frac{1}{\Delta^3}\delta\left(\frac{x-X_M}{\Delta}\right)\delta\left(\frac{y-Y_M}{\Delta}\right)\delta\left(\frac{z-Z_M}{\Delta}\right), \tag{17}$$

where

$$\delta(r)=\begin{cases}\frac{1}{6}(5-3|r|-\sqrt{-3(1-|r|)^2+1}, & 0.5\leqslant|r|\leqslant1.5,\quad r=\frac{(x-x_0)}{\Delta},\\ \frac{1}{3}(1+\sqrt{-3r^2+1}, & |r|\leqslant0.5,\\ 0, & \text{otherwise.}\end{cases}\tag{18}$$

The same interpolation kernel can be used to interpolate an Eulerian quantity defined at the grid centroids to the material volume centroids. The interpolation kernel is second-order accurate for smoothly varying fields [5]. Fig. 2 shows the effect of interpolation kernel applied to compute volume fraction for a spherical particle to show the effect of material volume refinement on particle boundary representation. The surface of the particle is smoothed over the scale proportional to the kernel length. Recently, Uhlmann [11] used similar interpolation scheme between the Lagrangian points and the background grid in his direct forcing immersed boundary technique. It was shown that the effect of increased stencil or the kernel width is to smooth the function being interpolated. Note that in order to reduce the spreading of the interfacial region, it is necessary to use compact support as well as finer background grids and material volumes.

3.3. Updating the particle position

The rigid body motion (RBM) of a particle can be decomposed into translational ($\mathbf{U}^T$) and rotational ($\mathbf{U}^R$) components The total velocity field at each point within the particle is given as

$$\mathbf{U}^{\mathrm{RBM}}=\mathbf{U}^T+\mathbf{\Omega}\times\mathbf{r},\tag{19}$$

where $\mathbf{U}^T$ is the translational velocity, $\mathbf{\Omega}$ the angular velocity, and $\mathbf{r}$ the position vector of the material volume centroid with respect to the particle centroid. All the material volumes have the same translational velocity as the particle centroid ($\mathbf{U}^T=\mathbf{U}_P$).

Given a velocity field and the positions ($\mathbf{X}_M^0$) of the material volume centroids and the particle centroid ($\mathbf{X}_P$) at $t=t_0$, the new positions ($\mathbf{X}_M^t$) at $t=t_0+\Delta t$ are obtained by linear superposition of the rotational and translational components of the velocity. The axis of rotation passing through the rigid body centroid $\mathbf{X}_P$ is given as $\hat{\boldsymbol{\sigma}}=\mathbf{\Omega}/|\mathbf{\Omega}|$. The new coordinates due to rotation around $\hat{\boldsymbol{\sigma}}$ are given as

$$\mathbf{X}'=\mathcal{R}(\mathbf{X}_M^0-\mathbf{X}_P)+\mathbf{X}_P,\tag{20}$$

where the rotation matrix is

$$\mathcal{R}=\begin{bmatrix}t\hat{\sigma}_x\hat{\sigma}_x+c & t\hat{\sigma}_x\hat{\sigma}_y-s\hat{\sigma}_z & t\hat{\sigma}_x\hat{\sigma}_z+s\hat{\sigma}_y\\ t\hat{\sigma}_x\hat{\sigma}_y+s\hat{\sigma}_z & t\hat{\sigma}_y\hat{\sigma}_y+c & t\hat{\sigma}_y\hat{\sigma}_z-s\hat{\sigma}_x\\ t\hat{\sigma}_x\hat{\sigma}_z-s\hat{\sigma}_y & t\hat{\sigma}_y\hat{\sigma}_z+s\hat{\sigma}_x & t\hat{\sigma}_z\hat{\sigma}_z+c\end{bmatrix}.\tag{21}$$

Here $c=\cos(\alpha)$, $s=\sin(\alpha)$, $t=1-\cos(\alpha)$, and $\alpha=|\mathbf{\Omega}|dt$. The material volume centroids are all uniformly translated to give the final positions,

$$\mathbf{X}_M^t=\mathbf{X}'+\mathbf{U}^T\,dt.\tag{22}$$

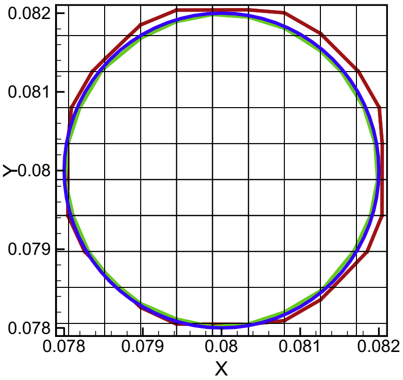


Fig. 2. Contour of particle volume fraction representing the surface of the particle and smoothing effect of the interpolation kernel: blue line is the actual particle boundary, red and green lines denote contours of volume fraction of 0.5 obtained from $\Delta/\Delta_M=1$ and 4, respectively.

3.4. Collision force modeling

As two or more particles come close to each other, a repulsive force strong enough to prevent overlapping of particle boundaries is necessary. A collision strategy similar to that used by Glowinski et al. [4] is used and is briefly described here for completeness. The repulsive force ($\mathbf{F}_P^{\text{coll}}$) on the body P due to Q is given as

$$\mathbf{F}_P^{\text{coll}} = \sum_{Q=1,Q\neq P}^{N_P} \mathbf{F}_{P,Q}^{\text{coll}} + \sum_{W=1}^{N_W} \mathbf{F}_{P,W}^{\text{coll}}. \tag{23}$$

It is a short-range repulsive force exerted on the Pth particle by other particles ($Q = 1, N_P$) and by nearby walls ($W = 1, N_W$).

For simplicity, consider two spherical rigid bodies (P and Q) of radius R_P and R_Q, respectively, undergoing collision. Let G_P and G_Q be their mass centers. The repulsive force ($\mathbf{F}_P^{\text{coll}}$) on the body P due to Q is given as

$$\mathbf{F}_{P,Q}^{\text{coll}} = \frac{\mathcal{C}_{P,Q}}{\epsilon}\left(\max\left\{0, -\left[\frac{\delta_{PQ} - R_P - R_Q - \mathcal{S}}{\mathcal{S}}\right]\right\}\right)^2 \frac{\overline{G_P G_Q}}{\delta_{PQ}}, \tag{24}$$

where $\delta_{PQ} = \|\overline{G_P G_Q}\|$, $\mathcal{S}$ is the range of the short-range repulsive force, $\overline{G_P G_Q}$ is the position vector between the centers of the two bodies, ϵ is a small positive number, $\mathcal{C}_{PQ}$ is the scaling factor and has the dimensions of a force [MLT^{-2}]. The range of repulsive force indicates the distance between the *boundaries* of the rigid bodies at which the repulsive force is activated. Typically, $\mathcal{S} = \Delta$, where Δ is the background grid resolution. The magnitude of the small positive number was determined using simple analysis of a falling sphere by Glowinski et al. [4]. Accordingly, $\epsilon \sim \Delta^2$ and $\mathcal{C}_{PQ} = Mg$ are used, where g is the gravitational acceleration, and M is the average mass of the particles P and Q.

The repulsive force between a particle P and wall is computed by generating a particle that is a mirror image of P (same size and at the same distance from the wall as P). The repulsive force is then computed between P and its mirror image and using the above formula. Note that the above collision strategy computes a repulsive force that is normal to the point of contact between the particles or particle and a wall. More sophisticated collision models involving shearing forces (tangential) and applicable to arbitrary shapes can be developed and used [22].

The collision force computation, could be very expensive ($\mathcal{O}(N_P^2)$ operations), if there are large number of particles, and the force is calculated by computing the inter-particle distances between each particle. The computational cost can be reduced owing to the material point representation of the particles. For each rigid particle P, a list of all material points that are on the boundary of the particle can be created. These material points then can be sorted according to the background grid cv(i, j, k) they belong to. Using the connectivity of the background grid, and finding the material points belonging to different rigid bodies, the collision calculation can be restricted to a few particles that are nearing actual collision. Advanced schemes involving Verlet lists [23] and linked-lists [24] can be used to reduce the computational overhead.

4. Discretization of the governing equations and numerical algorithm

Fig. 3 shows the schematic of variable storage in time and space. The particle-positions, density, pressure and volume fractions are staggered in time with respect to the fluid and particle velocity fields, u_i and U_i, respectively. All variables are stored at the control volume (cv) center with the exception of the face-normal velocity u_N, located at the face centers. The face-normal velocity is used to enforce continuity equation. Capital letters are used to denote particle fields. The time-staggering is done so that the variables are located most conveniently for the time-advancement scheme. We follow the collocated spatial arrangement for velocity and pressure field as has been used by [25,18,19]. The main reason to use this arrangement as opposed spatial-staggering is the flexibility of extending the scheme to unstructured grids and/or adaptive mesh refinement. In the present work, however, uniform Cartesian grids are used for simplicity. Accordingly, the particle positions (X_i), density (ρ), volume fraction (Θ), and viscosity (μ) are located at time level $t^{n+1/2}$ and $t^{n+3/2}$ whereas the velocity field (u_i, u_N, and U_i), the pressure (p), and the rigid body constraint force $f_{i,R}$ are located at time level t^n and t^{n+1}. This makes the discretization symmetric in time, a feature important to obtain good conservation properties of the numerical scheme as emphasized and used by Pierce and Moin [26] for low-Mach number, reactive flows.

Using these variable locations, integrating the governing equations over the control volume and applying Gauss' divergence theorem to transport volume integrals to surface integrals wherever possible, the discrete governing equations are derived. Accordingly, the continuity equation is

$$\frac{\rho_{\text{cv}}^{n+3/2} - \rho_{\text{cv}}^{n+1/2}}{\Delta t} + \frac{1}{V_{\text{cv}}} \sum_{\text{faces of cv}} \rho_{\text{face}}^{n+1} u_N^{n+1} A_{\text{face}} = 0, \tag{25}$$

where the subscript "face" corresponds to the face-center, A_{face} is the face area, and V_{cv} the volume of the control volume. The density field (at any discrete time) is a linear function of the volume fraction and is given as

$$\rho_{\text{cv}} = \rho_F \Theta_F + \rho_P \Theta_P, \tag{26}$$

where the subscripts "F" and "P" stand for fluid and particle, respectively. The density at the faces of a control volume (ρ_{face}) is obtained by doing simple arithmetic averages of the density at cvs adjacent to the face. In addition the volume fraction

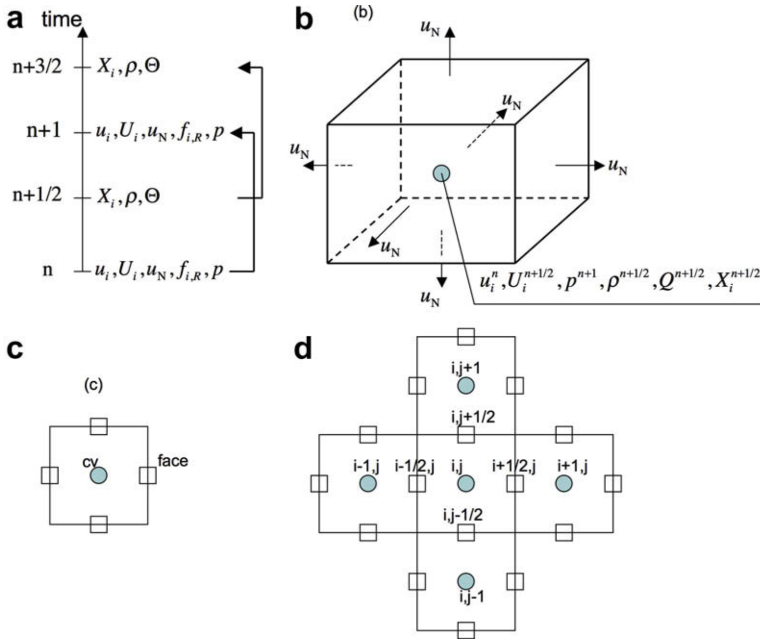


Fig. 3. Schematic of the variable storage in time and space: (a) time-staggering, (b) three-dimensional variable storage, (c) cv and face notation, (d) index notation. The velocity field (u_i, u_N), the pressure field (p), and the rigid body force ($f_{i,R}$) are staggered in time with respect to the volume fraction (Θ), density (ρ), and particle position (X_i). All variables are collocated in space at the centroid of a control volume except the face-normal velocity u_N which is stored at the centroid of the faces of the control volume.

fields follow the conservation relation $\Theta_P + \Theta_F = 1$. The particle volume fraction is a function of the position of the particle and is obtained by interpolation procedure described previously.

For the present work we assume that the fluid and particle densities are constant. However, the formulation is general and can be used to include variations in densities of each material due to chemical reactions or temperature variations.

In addition, the density field at time level (t^{n+1}) can be obtained by taking arithmetic averages of values at $t^{n+1/2}$ and $t^{n+3/2}$ or by directly evaluating it based on material volume locations at $X_{i,M}^{n+1}$.

The discrete momentum equation for the ith component of velocity is

$$\frac{g_{i,\mathrm{cv}}^{n+1} - g_{i,\mathrm{cv}}^{n}}{\Delta t} + \frac{1}{V_{\mathrm{cv}}} \sum_{\text{faces of } cv} g_{i,\mathrm{face}}^{n+1/2} u_N^{n+1/2} A_{\mathrm{face}} = -\frac{\partial}{\partial x_i} p_{\mathrm{cv}}^{n+1} + \frac{1}{V_{\mathrm{cv}}} \sum_{\text{faces of } cv} (\tau_{ij})_{\mathrm{face}}^{n+1/2} N_{j,\mathrm{face}} A_{\mathrm{face}} + f_{i,\mathrm{cv}}^{n+1}, \tag{27}$$

where $g_i = \rho u_i$ represents the momentum in the i^{th} direction, $(\tau_{ij})_{\mathrm{face}}$ is the viscous stress at the faces of control volume, and $N_{j,\mathrm{face}}$ represents the components of the outward face-normal. The velocity field ($u_{i,\mathrm{face}}$), and the momentum $\rho u_{i,\mathrm{face}}$, and the density (ρ_{face}) at the faces are obtained using arithmetic averages of the corresponding fields at two control volumes associated with the face. The values at time level $n + 1/2$ are obtained by time-averaging (Crank Nicholson). f_i represents the force due to the interphase coupling. This force is used to impose the rigidity constraint within the particle domain and can be obtained by determining the rigid body motion as described below.

Let $\mathbf{u}$ be the velocity field at the fixed grid points within the particle. The corresponding velocities at the material volumes within the particle be $\mathbf{U}_M$, which can be obtained by following the interpolation procedure described earlier. The interpolated velocity may not necessarily represent a pure rigid body motion. Hence, the velocity field at the material volumes can be further decomposed as

$$\mathbf{U}_M = \mathbf{U}_M^{\mathrm{RBM}} + \mathbf{U}'_M, \tag{28}$$

where $\mathbf{U}_M^{\mathrm{RBM}}$ is the rigid body motion and $\mathbf{U}'_M$ represents the remaining non-rigid motion satisfying the continuity equation. The rigid body motion consists of translational ($\mathbf{U}_M^T$) and rotational ($\mathbf{U}_M^R$) velocity components. The rotational component at each material point is related to the angular velocity of the particle, $\mathbf{U}_M^R = \mathbf{\Omega}_P \times \mathbf{r}$ whereas the translation component is the same as the velocity of the centroid of the particle. Here, $\mathbf{r}$ is the position vector of the material point "M" from the particle centroid. The rigid body motion can be obtained as

$$\mathcal{M}_P \mathbf{U}_P = \sum_{M=1}^{N} V_M \rho_M \mathbf{U}_M, \tag{29}$$

$$\mathcal{I}_P \mathbf{\Omega}_P = \sum_{M=1}^{N} \rho_M V_M (\mathbf{r} \times \mathbf{U}_M), \tag{30}$$

where subscripts P and M denote the particle and the material volume centroids, respectively, V_M is the volume and ρ_M the density of each material volume, $\mathcal{M}_P = \sum_{M=1}^{N}\rho_M V_M$ is the total mass of the particle, and $\mathcal{I}_P$ is the moment of inertia of the particle about the coordinate axes fixed to the particle centroid. The moment of inertia is given as

$$\mathcal{I}_P = \sum_{M=1}^{N} \rho_M V_M[(\mathbf{r}\cdot\mathbf{r})\mathbf{I} - \mathbf{r}\otimes\mathbf{r}], \tag{31}$$

where $\mathbf{I}$ represents the identity matrix.

The rigid-body constraint is satisfied by imposing a volumetric force on the fluid equations. The value of this force at the material volume centroids can be obtained as,

$$F_{i,M}^{n+1} = -\frac{\rho_M^{n+1}(U_{i,M} - U_{i,M}^{\text{RBM}})}{\Delta t}. \tag{32}$$

The force on the grid control volumes ($f_{i,\text{cv}}$) can be obtained from $F_{i,M}$ by using the same interpolation scheme discussed earlier (Eq. 14). The actual implementation of the formulation requires two-stage, fractional time-stepping wherein the continuity and rigidity constraints on the velocity field are imposed in different fractional steps. The numerical algorithm is discussed below.

A semi-implicit numerical scheme with an iterative approach is given below. In the following steps, we are advancing the particle positions from time level $t^{n+1/2}$ to $t^{n+3/2}$ and the velocity fields from t^n to t^{n+1}. The superscript k refers to iteration cycles between the respective time levels. Note that the algorithm is designed to allow for multiple iterations, however, it was found that, at small time steps with CFL $\leqslant$ 0.5, a single iteration is enough to obtain accurate and stable results. We use the algebraic multigrid approach for the Poisson equation based on the *Hypre* library from Lawrence Livermore National Laboratory [27], making multiple iterations per time-step feasible.

1. **Choose predictors at** $k = 0$
   Choose predictors (initial guesses) for the values of the variables at the next time level. We first choose the velocity predictors using the Adams–Bashforth extrapolation:

$$\left.\begin{aligned} u_N^{n+1,0} &= 2u_N^n - u_N^{n-1};\, u_{i,\text{cv}}^{n+1,0} = 2u_{i,\text{cv}}^n - u_{i,\text{cv}}^{n-1}, \\ U_{i,P}^{n+1,0} &= 2U_{i,P}^n - U_{i,P}^{n-1};\, \Omega_{i,P}^{n+1,0} = 2\Omega_{i,P}^n - \Omega_{i,P}^{n-1}. \end{aligned}\right\} \tag{33}$$

   The pressure is updated using backward Euler, $p_{\text{cv}}^{n+1,0} = p_{\text{cv}}^n$. Likewise the rigidity constraint ($f_{i,\text{cv}}^{n+1,0}$) is obtained by performing interpolations from material volume locations at $X_{i,M}^{n+1}$ and using the predictors $u_{i,\text{cv}}^n$, $U_{i,P}^n$.
2. **Update particle positions and compute scalars (volume fraction and density fields)**
   We first advance the particle positions from $t^{n+1/2}$ to t^{n+1} (half the time-steps) using the predictor velocities at t^{n+1}. The particle positions will be corrected later.

$$X_{i,M}^{n+1} = X_{i,P}^{n+1/2} + \mathcal{R}_{ij}\left(X_{j,M}^{n+1/2} - X_{j,P}^{n+1/2}\right) + U_{i,M}^{n+1,0}\frac{\Delta t}{2}, \tag{34}$$

   where $\mathcal{R}_{ij}$ is evaluated from Eq. (21) and using particle locations at $t^{n+1/2}$, $\Omega_{i,M}^{n+1,0}$, and $\Delta t/2$ Once the new positions are known, compute the scalar fields, Θ_P^{n+1} and the corresponding ρ^{n+1} using Eq. (26). The temporal change in density is set equal to:

$$\frac{\delta\rho_{\text{cv}}^{n+1}}{\delta t} = \frac{\rho_{\text{cv}}^{n+1} - \rho_{\text{cv}}^{n+1/2}}{\Delta t/2}. \tag{35}$$

3. **Start iteration** $k+1$. **Advance the momentum equations using the fractional step method.**
   We advance the velocity field from t^n to $t^{n+1,k+1}$ in few iterations. The intermediate velocity fields may not satisfy the continuity or the rigidity constraints. These are enforced later.

$$\frac{\rho_{\text{cv}}^{n+1}\widehat{u}_{i,\text{cv}}^{k+1} - \rho_{\text{cv}}^n u_{i,\text{cv}}^n}{\Delta t} + \frac{1}{V_{\text{cv}}}\sum_{\text{faces of cv}} \widehat{g}_{i,\text{face}}^{k+1/2} u_N^{k+1/2} A_{\text{face}} = -\frac{\partial}{\partial x_i}(p_{\text{cv}}^{n+1,k}) + \frac{1}{V_{\text{cv}}}\sum_{\text{face of cv}} \widehat{\tau}_{ij,\text{face}}^{k+1/2} N_{j,\text{face}} A_{\text{face}} + f_{i,\text{cv}}^{n+1,k}, \tag{36}$$

   where

$$\begin{aligned} \widehat{g}_{i,\text{face}}^{k+1/2} &= \frac{1}{2}(g_{i,\text{face}}^n + \widehat{g}_{i,\text{face}}^{k+1}), \\ \widehat{\tau}_{ij,\text{face}}^{k+1/2} &= \mu_F\left[\frac{1}{2}\left(\frac{\partial u_i^n}{\partial x_j} + \frac{\partial \widehat{u}_i^{k+1}}{\partial x_j}\right) + \frac{1}{2}\left(\frac{\partial u_j^n}{\partial x_i} + \frac{\partial \widehat{u}_j^{k}}{\partial x_i}\right)\right]_{\text{face}}, \\ u_N^{k+1/2} &= \frac{1}{2}\left(u_N^n + u_N^{n+1,k}\right). \end{aligned}$$

The convective terms and the viscous stresses in the direction of the momentum g_i are treated implicitly. For the face-normal velocities, the latest available value of u_N is used in the above equation. Note that $f_{i,\text{cv}}^{n+1,k}$ is obtained by interpolation from the material volume positions at $X_{i,M}^{n+1}$.

4. **Remove the old pressure gradient**

$$\widehat{\widehat{g}}_{i,\text{cv}}^{k+1} = \widehat{g}_{i,\text{cv}}^{k+1} + \Delta t \frac{\partial}{\partial x_i}(p_{\text{cv}}^{n+1,k}). \tag{37}$$

5. **Interpolate the** cv-**center momentum** (**velocities**) **to faces to obtain face-normal momentum** (**velocity**)

$$\widehat{\widehat{g}}_N^{k+1} = \frac{1}{2}(\widehat{\widehat{g}}_{i,\text{icv1}}^{k+1} + \widehat{\widehat{g}}_{i,\text{icv2}}^{k+1})N_{i,\text{face}}, \tag{38}$$

where for face $[i,j+1/2]$, the neighboring cvs correspond to $[i,j]$ (icv1) and $[i,j+1]$ (icv2), respectively.

6. **Solve the pressure equation**

$$\sum_{\text{faces of cv}} \frac{\delta}{\delta N}(p_{\text{cv}}^{n+1,k+1})A_{\text{face}} = \frac{1}{\Delta t}\sum_{\text{face of cv}} \widehat{\widehat{g}}_N^{k+1} A_{\text{face}} + V_{\text{cv}}\frac{\delta \rho_{\text{cv}}^{n+1}}{\delta t}. \tag{39}$$

7. **Update the face-normal velocities to new continuity-satisfying field**

$$u_N^{n+1,k+1} = \frac{1}{\rho_{\text{face}}^{n+1}}\left(\widehat{\widehat{g}}_N^{k+1} - \Delta t \frac{\delta}{\delta N} p_{\text{cv}}^{n+1,k+1}\right). \tag{40}$$

Note that this implies that the face-normal velocities are obtained from pressure-projection and are not interpolated from the adjacent control volumes, thus ensuring strong coupling between the pressure-gradient and the velocity field.

8. **Reconstruct the pressure gradient and cv-center velocities**

$$\frac{\delta p^{n+1,k+1}}{\delta x_i} = \overline{\left(\frac{\delta p^{n+1,k+1}}{\delta N}\right)}^{\text{face}\rightarrow\text{cv}}, \tag{41}$$

where $\overline{()}^{\text{face}\rightarrow cv}$ stands for reconstruction of the pressure gradient at the cv-centers from the corresponding face-normal gradients. For example, the x-component of the cell-centered pressure gradient $\frac{\delta p}{\delta x}$ is obtained as:

$$\frac{\delta p}{\delta x} = \frac{\sum_{\text{faces of cv}} \frac{\delta p}{\delta N}\cdot \vec{i}\|N_{i,\text{face}}A_{\text{face}}\|}{\sum_{\text{faces of cv}}\|N_{i,\text{face}}A_{\text{face}}\|}. \tag{42}$$

For non-uniform and unstructured grids, a least-squares based face area-weighted interpolation was first proposed by Mahesh et al. [18]. For uniform Cartesian grids, it is equivalent to the above reconstruction. In this work, we use Cartesian uniform grids, however, the numerical scheme can be extended to more complex grids using the least-squares area-weighted reconstruction.

$$u_{i,\text{cv}}^{*,k+1} = \frac{1}{\rho_{\text{cv}}^{n+1}}\left(\widehat{\widehat{g}}_{i,\text{cv}}^{k+1} - \Delta t \frac{\delta}{\delta x_i} p_{\text{cv}}^{n+1,k+1}\right). \tag{43}$$

Note that in the absence of a rigid body, $\rho = \rho_F$ throughout the domain, and the algorithm reduces to the standard fractional step scheme for single-phase, incompressible flow. The above velocity field will then be denoted as $u_{i,\text{cv}}^{n+1,k+1}$ and outer iterations can be continued till convergence. In the presence of rigid bodies, the following steps are performed to enforce the rigidity constraint within the particle domain.

9. **Remove the old rigidity constraint force**

$$u_{i,\text{cv}}^{**,k+1} = \frac{1}{\rho_{\text{cv}}^{n+1}}\left(u_{i,\text{cv}}^{*,k+1} - \Delta t f_{i,\text{cv}}^{n+1,k}\right). \tag{44}$$

10. **Compute the rigid-body motion**
First interpolate the velocity field $u_{i,\text{cv}}^{**,k+1}$ from the grid cvs to the material volume centroids to obtain $U_{i,M}^{**,k+1}$. Solve for the translational and rotational velocity fields using Eq. 29 and compute the rigid body motion:

$$\mathbf{U}_M^{\text{RBM},k+1} = \mathbf{U}_M^{T,n+1,k+1} + \mathbf{\Omega}_P^{n+1,k+1} \times (\mathbf{X}_M^{n+1} - \mathbf{X}_P^{n+1}). \tag{45}$$

11. **Compute the rigidity constraint force**
First compute the rigid-body constraint force at the material volume centroids.

$$F_{i,M}^{n+1,k+1} = \rho_M^{n+1}\frac{U_{i,M}^{**,k+1} - U_{i,M}^{\text{RBM},k+1}}{\Delta t}. \tag{46}$$

Interpolate the rigidity constraint force to the grid control volumes to obtain $f_{i,\text{cv}}^{n+1,k+1}$.

12. **Enforce the rigidity constraint force**

$$u_{i,\mathrm{cv}}^{n+1,k+1} = u_{i,\mathrm{cv}}^{**,k+1} + \frac{\Delta t}{\rho_{\mathrm{cv}}^{n+1}}\left(f_{i,\mathrm{cv}}^{n+1,k+1}\right). \tag{47}$$

13. **Check for convergence and repeat**
Set k to $k+1$ and go to step 3. We check for convergence in the velocity field for each iteration by evaluating change in velocity across iterations to find whether convergence is achieved for each time-step. Typically, with the good predictor guess convergence can be achieved in 2–3 iterations.
14. **Reset the particle positions and velocities**

$$U_{i,M}^{n+1} = U_{i,M}^{T,n+1,k+1}, \tag{48}$$

$$\Omega_{i,M}^{n+1} = \Omega_{i,M}^{n+1,k+1}, \tag{49}$$

$$X_{i,M}^{n+3/2} = X_{i,P}^{n+1/2} + \mathcal{R}_{ij}(X_{j,M}^{n+1/2} - X_{j,P}^{n+1/2}) + U_{i,M}^{T,n+1}\Delta t, \tag{50}$$

where $\mathcal{R}_{ij}$ is evaluated from Eq. 21 and using particle locations at $t^{n+1/2}$, $\Omega_{i,m}^{n+1}$, and Δt

4.0.1. Numerical errors and accuracy

From the above steps, it can be shown that the total splitting error in the above fractional step is:

$$u_{i,\mathrm{cv}}^{n+1} - \hat{u}_{i,\mathrm{cv}}^{k+1} = -\Delta t\left[\frac{\delta}{\delta x_i}(p_{\mathrm{cv}}^{n+1,k+1} - p_{\mathrm{cv}}^{n+1,k}) - \left(f_{i,\mathrm{cv}}^{n+1,k+1} - f_{i,\mathrm{cv}}^{n+1,k}\right)\right] \tag{51}$$

$$= -\Delta t^2\left[\frac{\delta}{\delta x_i}\left(\frac{\delta p_{\mathrm{cv}}}{\delta t}\right) + \frac{\delta f_{i,\mathrm{cv}}}{\delta t}\right] + \mathcal{O}\left(\delta p_{\mathrm{cv}}^2 + \delta f_{i,\mathrm{cv}}^2\right). \tag{52}$$

Note that δp_{cv} and $\delta f_{i,\mathrm{cv}}$ are defined as the differences between iteration levels for the pressure and the rigidity constraint force, respectively. By performing multiple iterations within each time step, the splitting error can be reduced. If $k = 1$, the splitting error is dependent on the initial guess for p_{cv}^{n+1} and $f_{i,\mathrm{icv}}^{n+1}$. The computational time for each subsequent iteration reduces significantly and hence the cost of multiple iterations is not significant. Typically 3–5 iterations are sufficient. In the present work, the time-step used is such that the maximum CFL $\leqslant 0.5$ at all times and only a single iteration is used.

The other source of numerical error is the interpolation from the grid points to the material volumes and back from the material volumes to the grid points. In the above algorithm, interpolations to the material volumes and back to the grid points are required once per time-step.

In the absence of a rigid body, the above algorithm consistently reduces to an incompressible flow, co-located scheme.

5. Numerical examples

We conduct a series of numerical studies to evaluate the accuracy of the scheme and its applicability to turbulent flows with large number of particles.

5.1. Decaying Taylor vortex

We first examine the accuracy of interpolation operator to transfer the flow quantities defined at the material volumes to the background grid cv-centers. The problem of stationary, decaying vortices in a periodic box is used to study the accuracy of the numerical scheme. The temporal and spatial solution for the velocity and pressure fields is given as

$$\left.\begin{aligned} u(x,y,t) &= -\cos(k_x x)\sin(k_y y)e^{-\mu(k_x^2+k_y^2)t},\\ v(x,y,t) &= \sin(k_x x)\cos(k_y y)e^{-\mu(k_x^2+k_y^2)t},\\ p(x,y,t) &= -0.25(\cos\left[(2k_x x) + \cos(2k_y y)\right]e^{-2\mu(k_x^2+k_y^2)t}, \end{aligned}\right\} \tag{53}$$

where $k_x = 2\pi/L_x$ and $k_y = 2\pi/L_y$. The domain size is $L_x \times L_y = 2 \times 2$ in non-dimensional units. A fictitious, square region of length 0.5 units is placed at the center of the domain. Material volumes are generated within the boundary by following the procedure described in Section 3. Initially, the immersed boundary is oriented such that its borders and the background mesh do not coincide as shown in Fig. 4.

The goal here is to investigate the effect of interpolation errors between the material volumes and the background mesh. The analytical solution for pressure and velocity is used as an initial condition at every grid point outside the square boundary. In addition, the fluid velocity and pressure fields at the material volume centers inside the square region are specified using the analytical solution at each time-step. This specified velocity field is projected onto the grid at each time step by using the interpolation procedure described in Section 3. Note that for this test case, the region inside the immersed boundary is *not* constrained to undergo a rigid body motion. Instead, the error in interpolation between the material volumes, that

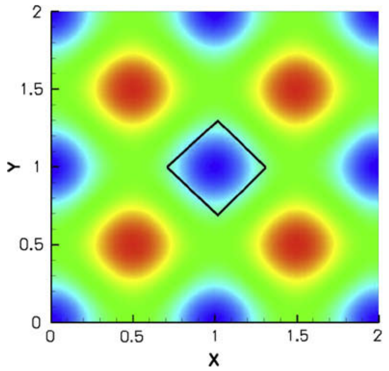


Fig. 4. Instantaneous pressure contours and orientation of the immersed boundary for the Taylor vortex case. The domain size is 2 × 2 units and the immersed boundary is a square object of length 0.5 units.

may not coincide with the grid cv-centers, and the cv-centers is examined. Periodic boundary conditions are applied at the boundaries of the domain. With this input the solution is integrated in time and compared with the analytical solution by computing L_∞ error of the velocity field and pressure fields at $t = 0.2$. The error in space and time is evaluated by reducing the grid size and time steps simultaneously. Computations were performed for uniform square grids with ($N = 10, 20, 40$ and 80) grid volumes for the background mesh. Constant time-steps are used and the ratio $\frac{\Delta t}{\Delta} = 0.1$ is held fixed. For each background mesh, the material volume resolution is also refined by keeping the ratio $\frac{\Delta}{\Delta_M} = 3$. The Reynolds number is set to 10 ($\mu = 0.1$ units). The density of the fluid and the material inside the immersed boundary are set to be unity. Thus, in the absence of any interpolation errors, the temporal and spatial evolution of the pressure and velocity field with or without the immersed object should be the same as the analytical solution. Three cases are investigated:

1. Absence of immersed boundary: In this case, the interpolation errors between the material volumes and the cv-centers are absent, and investigates the accuracy of the basic flow solver.
2. Fixed immersed boundary: A square region of size $L_{\rm sq} = 0.5$ is placed at the center of the domain. The square region contains material volumes of size $\frac{\Delta}{\Delta_M} = 3$, where Δ is the size of the background grid and Δ_M is the size of the material element. At each time-step, the analytical velocity field is specified at each material element, and is interpolated to the background cv-centers.
3. Rotated immersed boundary: In this case, in addition to placing a square region at the center of the domain, the square region and corresponding material volumes are rotated anti-clockwise around the center of the domain at each time-step. The only difference between this case and the one above is that the region occupied by the material volumes is changing at each time-step. Thus, the region to which interpolation of the specified velocity (at material volume centers) and the background cv-centers is applied, is changing in time. The period of rotation (T) is set to be 0.2 giving the rate of rotation $\omega = \frac{2\pi}{T} = 10\pi$.

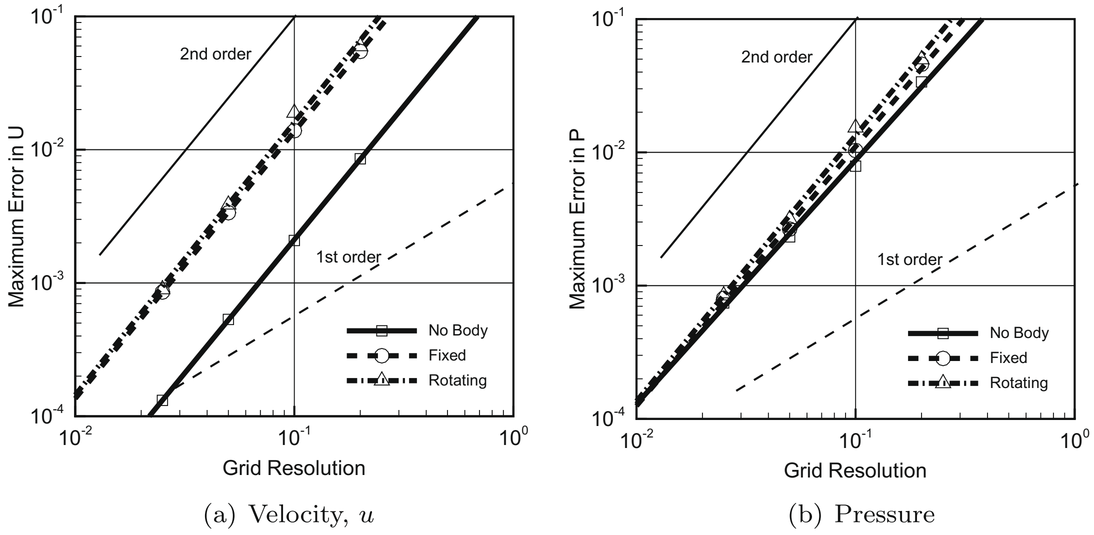
Fig. 5 shows the L_∞ error in the axial velocity (u) and the pressure. For the specified velocities at the material volume centroids, this tests the accuracy of the interpolation scheme and also the effect of the embedded body on the overall accuracy of the scheme. With the presence of the immersed boundary (stationary or rotating), the interpolation between the material volumes and the background mesh, results in an order of magnitude increase in the velocity error, however, the error converges with second-order accuracy. Similar behavior for the error in pressure is observed.

5.2. Externally forced motion of particles

In order to investigate the effectiveness of the scheme in capturing the fluid flow over immersed objects which are either fixed or forced to move in a specific manner, two problems are considered: (a) flow over a fixed cylinder, (b) flow over a fixed sphere, and (c) flow induced by inline oscillation of a circular cylinder.

5.2.1. Flow over a fixed cylinder

We first perform simulations of flow past a fixed circular cylinder at Reynolds numbers of 40, 100, 300, and 1000 and compare the results with available numerical and experimental data. The computational domain is $40D_p \times 40D_p$ in the x and y directions, respectively. We use two grid points in the z direction with periodic boundary conditions. The domain size in the z direction is such that we obtain cubic grid elements in the region of the cylinder. The material volume resolution is set based on the ratio $\frac{\Delta}{\Delta_M} = 4$. Note that, in this study, the particle is represented by cubic material volumes with stair-stepped representation of the boundary (Section 3). Uniform flow of $U_\infty = 1$ units is imposed at the left boundary of the domain. A convective outflow boundary condition is imposed at the exit. Slip condition ($\frac{\partial u}{\partial N} = 0$; where N represents normal to



(a) Velocity, u

(b) Pressure

Fig. 5. The L_∞ error in u and p at $t = 0.2$ for three cases: without the immersed boundary, with a fixed immersed boundary, with a rotating immersed boundary. The motion inside the immersed boundary is specified using the analytical solution for fluid velocity at the material volumes.

the boundary), is imposed on the boundaries in the vertical and spanwise directions. The fluid viscosity is varied to simulate flow over a sphere at different Reynolds numbers. Since the sphere is fixed, the material volumes are assigned a velocity of $U_{i,m} = 0$, $\Omega_{i,m} = 0$ and setting $U_{i,m}^{\text{RBM},k+1} = 0$ in evaluating the rigidity constraint [i.e. step (11)] of the numerical algorithm described in Section 4.

Three grid resolutions are employed to study the grid convergence effects: (i) coarse grid, 350 × 350 with 35 grid points inside the cylinder, (ii) medium grid, 500 × 500 with 60 grid points inside the cylinder, and (iii) fine grid, 600 × 600 with 100 grid points inside the cylinder region (see Fig. 6). The mesh is refined near the cylinder boundary. For comparison, Marella et al. [28] used a Cartesian grid method, and employed 452 × 452 mesh on a $30D_p \times 30D_p$ domain for similar test case, whereas, Mittal et al. [29] used 417 × 289 grid points on a $40D_p \times 40D_p$ domain. The grid sizes near the cylinder interface were $0.01D_p$, same as in the present fine-grid case.

Fig. 7 shows instantaneous spanwise vorticity contours for $Re_p = 100$, 300, and 1000 showing periodic Karman vortex shedding. Fig. 8 shows the temporal evolution of the drag ($C_D = F_D/\left(\frac{1}{2}\rho U_\infty^2 D_p L_z\right)$ and lift ($C_D = F_D/\left(\frac{1}{2}\rho U_\infty^2 D_p L_z\right)$ coefficients for the same Reynolds numbers, where L_z is the length in the z-direction. The temporal evolution for drag and lift coefficients and in good agreement with those reported by Mittal et al. [29]. The flow remains symmetric and steady at $Re_p = 40$ resulting in a steady drag.

In order to validate the accuracy of these computations, we compare the time averaged mean and *rms* velocity profiles as well as the Reynolds stress profiles in the wake regions with those obtained from a body-fitted grid solver [19] at grid

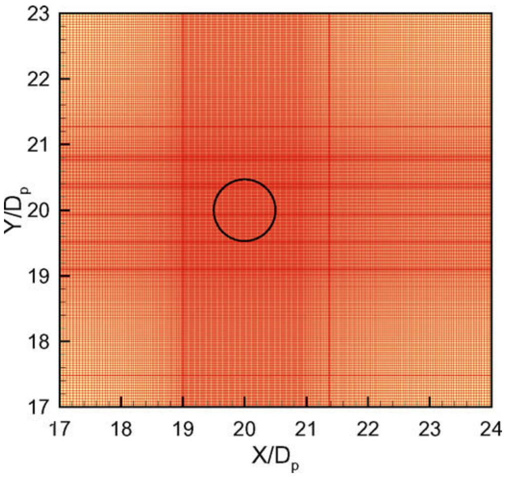
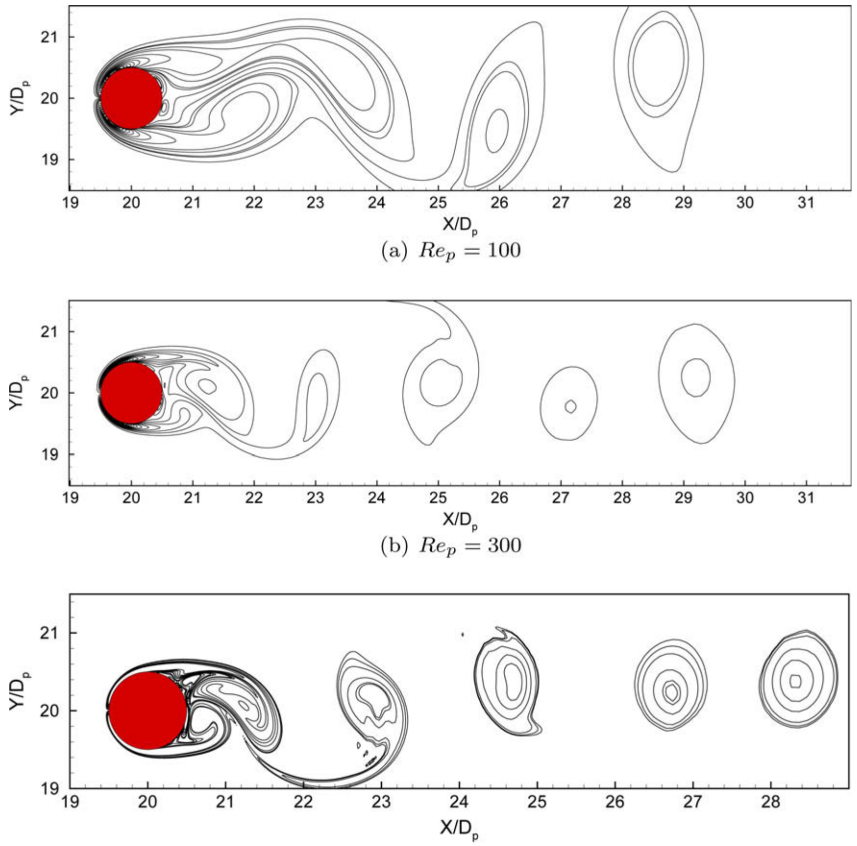


Fig. 6. Close-up view of the computational grid (fine resolution) used for flow over a cylinder.



(a) $Re_p = 100$

(b) $Re_p = 300$

(c) $Re_p = 1000$

Fig. 7. Instantaneous out-of plane vorticity contours for flow over a fixed cylinder at different Reynolds numbers showing Karman vortex shedding.

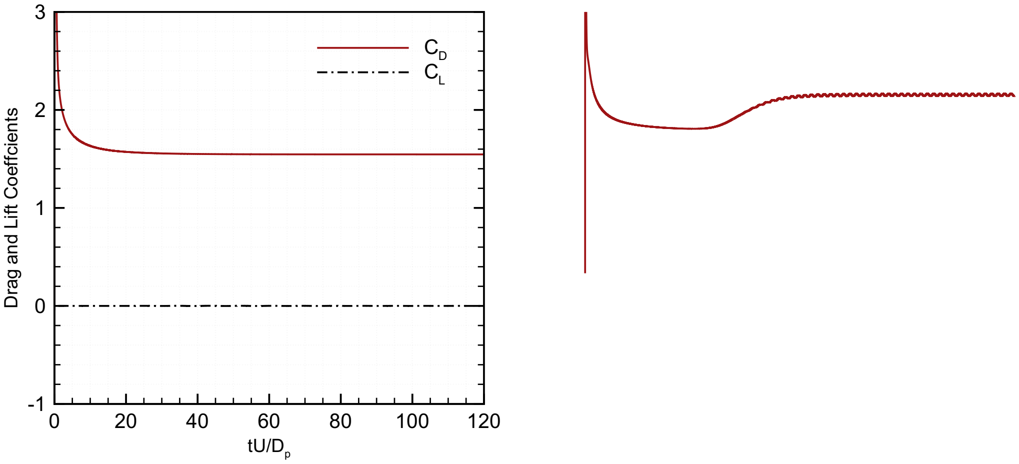
resolutions near the cylinder boundary similar to those employed in the present fictitious-domain calculations. Fig. 9 compares these wake statistics, showing very good agreement with the body-fitted grid solution.

In order to further validate the simulations, we have computed the mean drag coefficient and the Strouhal numbers ($St = fD_p/U_\infty$), where f is the vortex shedding frequency computed from the variation in the lift coefficient. These quantities are computed after a stationary state has been reached in each case. Table 1 shows the comparison of the Strouhal numbers with other studies, whereas Table 2 shows corresponding mean drag coefficients. The Strouhal numbers, mean drag coefficients, as well as the temporal evolutions of drag and lift coefficients are well predicted by the present scheme.

5.2.2. Flow over a fixed sphere

We perform three-dimensional simulations of flow over a fixed sphere in a uniform stream to investigate the accuracy of the numerical scheme to predict the drag coefficient and wake effects at different Reynolds numbers. Similar to the cylinder case, a domain of size $15D_p \times 15D_p \times 15D_p$ is used in the present study with inflow and convective outflow conditions in the x directions and slip-conditions in the y and z directions. The computational grid consists of 128^3 elements with uniform cubic elements in a small patch of $1.5D_p \times 1.5D_p \times 1.5D_p$ around the sphere. This gives around 26 grid points with in the spherical region. For comparison, Mittal et al. [29] used a domain of $16D_p \times 15D_p \times 15D_p$ with a non-uniform grid of $192 \times 120 \times 120$ grid points for $Re_p = 350$. In addition, we have performed coarse grid simulations (with around 10 grid points within the sphere) for comparison.

Fig. 10 shows streamlines in the symmetry plane over the fixed sphere at different Reynolds numbers. The flow remains symmetric for low Reynolds numbers, whereas vortex shedding is observed at large Re_p. For low Re_p, the location of the center of the recirculation bubble (x_c and y_c) and its length (L_b) were obtained and compared with published data to show good agreement (see Table 3). Similarly, Table 4 compares the mean drag coefficients with other studies for the fine grid resolutions to show good quantitative predictions. Fig. 11 shows this comparison with data from Clift et al. [38] and other studies. Predictions on *coarse grids* (with only 10 grid points within the sphere) also show similar trends; however, errors are in the range of 2% for large Reynolds numbers ($Re_p = 350$).



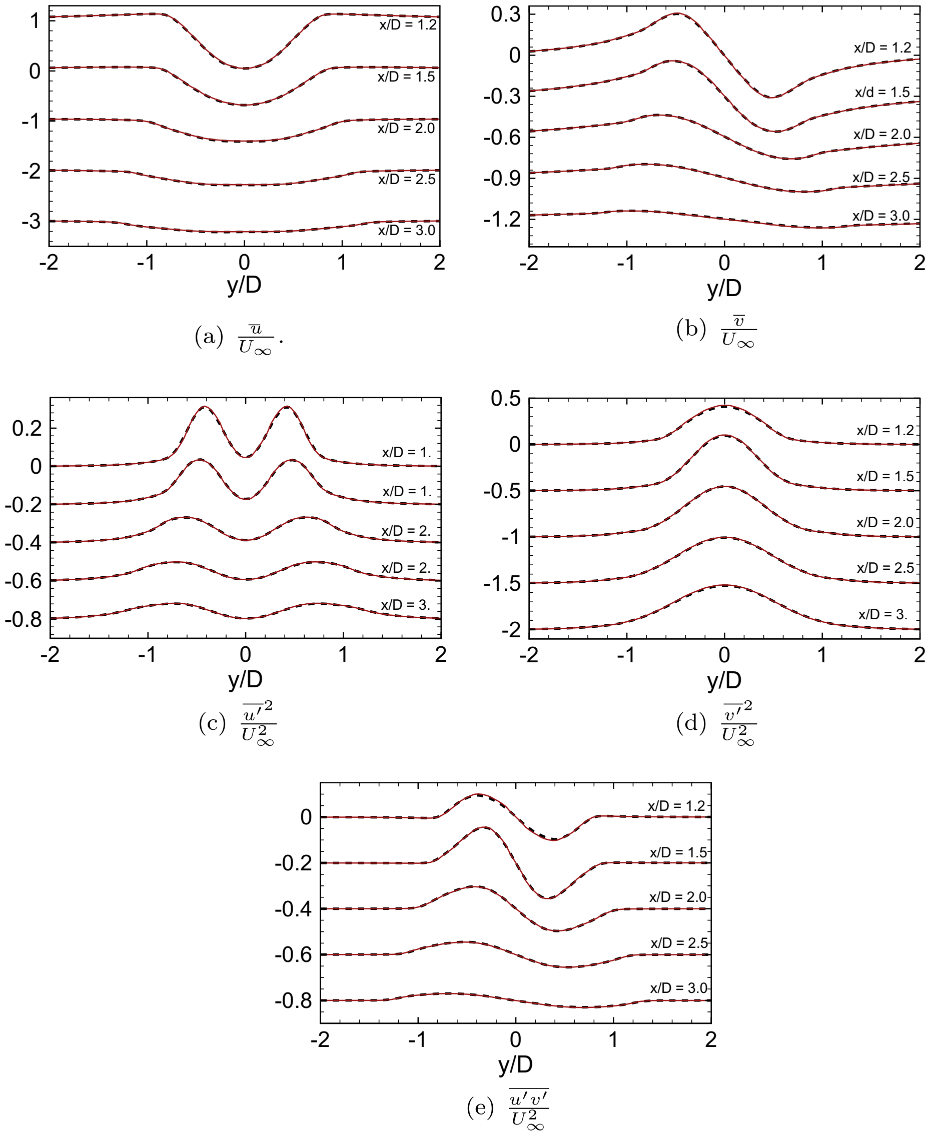
(a) $Re_p = 40$

5.3. Flow induced by inline oscillation of a circular cylinder

Flow induced by periodically oscillating circular cylinder was investigated in detail using experimental and numerical approaches by Dütsch et al. [41]. The flow is characterized by the Reynolds number $Re = \frac{\rho U_m D_P}{\mu}$ and the Keulegan–Carpenter number $KC = \frac{U_m f}{D_P}$ where U_m is the maximum velocity of the cylinder during oscillation, D_P is the cylinder diameter, ρ and μ are the fluid density and dynamic viscosity, respectively, f is the frequency of oscillation. The Keulegan–Carpenter number is basically inverse of a Strouhal number, except that the frequency of oscillation is based on the cylinder motion. In this test case, a sinusoidally varying translational velocity is imposed to the cylinder in the x-direction:

$$x_P(t) = -A_P \sin(\omega t), \tag{54}$$

where x_P is the x-location of the cylinder centroid, A_p is the amplitude of the oscillation, and $\omega = 2\pi f$ and the Keulegan–Carpenter number becomes $KC = 2\pi A_P/D_P$. The experiments by Dütsch et al. [41] were performed at $Re = 100$ and $KC = 5$. Accordingly, we use $D_P = 0.01$ m, $f = 5$ Hz, $U_m = 0.01$ m/s, $\rho = 1000$ kg/m^3, and $\mu = 1.0082 \times 10^{-3}$ kg/ms. Since we are interested in the fluid motion induced under *forced* oscillations of the cylinder, the density of cylinder is not necessary. The computational domain is a rectangular box of cross-section $100D_P \times 100D_P \times D_P$ in the axial, vertical, and spanwise directions. The cylinder is initially placed at the center of the box. Note that our computations are three-dimensional, the domain length is equal to one diameter in the spanwise direction and we apply periodic boundary conditions. For the x and y boundaries Neumann boundary condition $\frac{\partial u_i}{\partial N_j} = 0$ is used where N_j is the normal vector to the boundary faces. We use uniform Cartesian grids in a square region of $20D_P \times 20D_P \times D_P$, that covers the minimum and maximum displacement of the cylinder. The grids are stretched away from this region. The grid resolution in the square region around the cylinder is $\frac{\Delta}{D_P} = 20$, whereas outside this region the resolution is $\frac{\Delta}{D_P} = 5$. The material volume grid resolution is fixed at $\frac{\Delta}{\Delta_M} = 4$.



(a) $\frac{\overline{u}}{U_\infty}$.

(b) $\frac{\overline{v}}{U_\infty}$

(c) $\frac{\overline{u'^2}}{U_\infty^2}$

(d) $\frac{\overline{v'^2}}{U_\infty^2}$

(e) $\frac{\overline{u'v'}}{U_\infty^2}$

Fig. 9. Comparison of wake statistics between the present scheme and a body-fitted grid solver based on the formulation by Mahesh et al. [18,19]: —— present scheme, - - - - - - body-fitted grid solution.

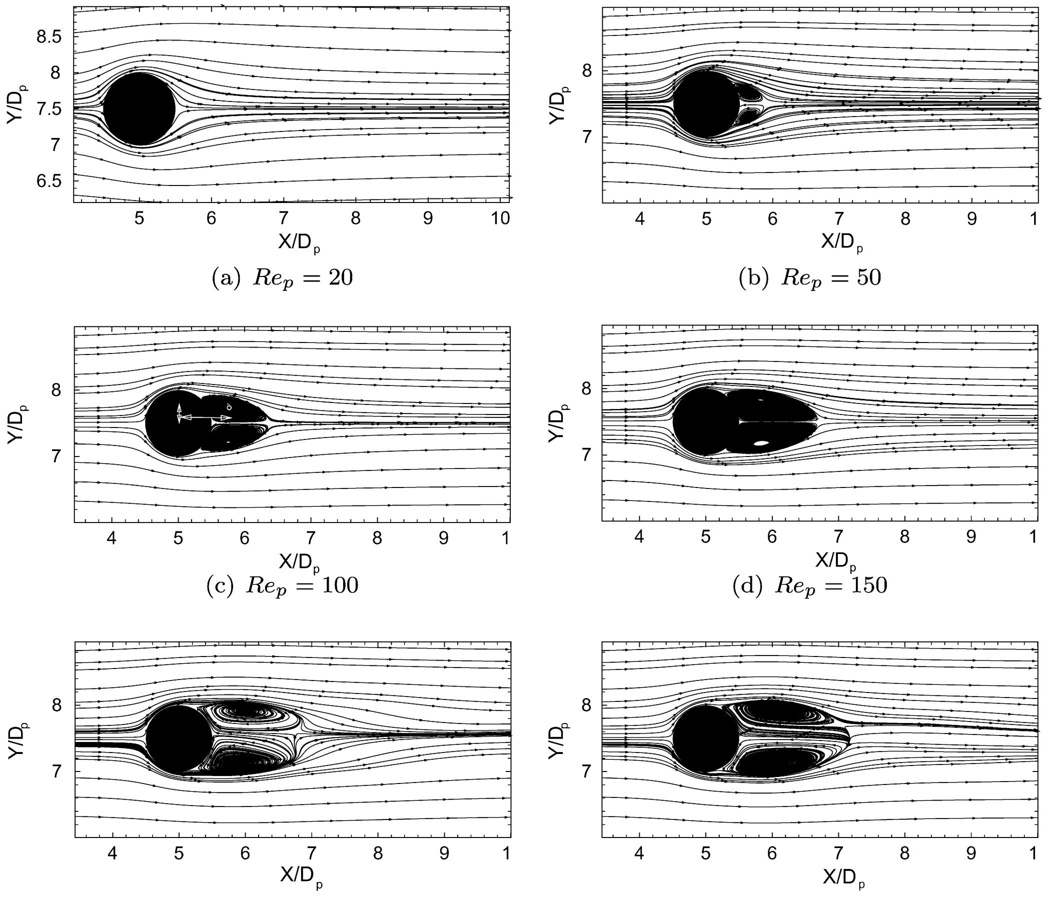
Table 1
Strouhal numbers for flow over a cylinder at different Reynolds numbers.

Study	Mesh	St_p		
		100	300	1000
Present scheme	Coarse	0.166	0.205	–
	Medium	0.165	0.211	–
	Fine	0.165	0.212	0.238
Williamson [30]		0.165	0.205	–
Mittal et al. [29]		0.165	0.21	0.231
Zhang et al. [31]		0.167	–	–

Fig. 12 shows the comparison of the normalized axial velocity in the vertical direction at three different cylinder positions. Our results are compared with the experimental data [41] as well as the computational results obtained by an immersed boundary method of Kim and Choi [12]. The fluid velocity induced by oscillatory motion of the cylinder is well predicted by our numerical scheme. Fig. 13 shows the normalized spanwise vorticity at different phase angles. The flow field developed

Table 2
Mean drag coefficient C_D for flow over a cylinder at different Reynolds numbers.

Study	Mesh	C_D			
		40	100	300	1000
Present scheme	Coarse	1.54	1.38	1.44	–
	Medium	1.53	1.37	1.42	–
	Fine	1.54	1.36	1.41	1.50
Henderson [32]		1.54	1.35	1.37	1.51
Mittal et al. [29]		1.53	1.35	1.36	1.45
Marella et al. [28]		1.52	1.36	1.28	–
Mittal and Balachandar [33]		–	–	1.37	–
Shu et al. [34]		–	1.38	–	–
Ye et al. [35]		1.52	–	1.38	–



(a) $Re_p = 20$ (b) $Re_p = 50$

(c) $Re_p = 100$ (d) $Re_p = 150$

(e) $Re_p = 300$ (f) $Re_p = 350$

Fig. 10. Instantaneous streamlines for flow over a fixed sphere at different Reynolds numbers.

Table 3
Comparison of key computed results for flow past a sphere with other experimental and numerical studies.

Re_d	50			100			150		
	x_c/D_p	y_c/D_p	L_b/D_p	x_c/D_p	y_c/D_p	L_b/D_p	x_c/D_p	y_c/D_p	L_b/D_p
Present scheme	0.617	0.204	0.382	0.757	0.287	0.866	0.324	0.32	1.2
Mittal et al. [29]	–	–	–	0.742	0.278	0.84	0.31	0.3	1.17
Marella et al. [28]	–	–	0.39	–	–	0.88	–	–	1.19
Johnson and Patel [36]	–	–	0.40	0.75	0.29	0.88	0.32	0.29	1.2
Taneda [37]	–	–	–	0.745	0.28	0.8	0.32	0.29	1.2

Table 4
Mean drag coefficient C_D for flow over a sphere at different Reynolds number.

Study	C_D					
Re_p	20	50	100	150	300	350
Present scheme	2.62	1.55	1.10	0.90	0.686	0.649
Mittal [39]	–	1.57	1.09	–	–	0.62
Mittal et al. [29]	–	–	1.08	0.88	0.68	0.63
Clift et al. [38]	2.61	1.57	1.09	0.89	0.684	0.644
Johnson and Patel [36]	–	1.57	1.08	0.9	0.629	–
Marella et al. [28]	–	1.56	1.06	0.85	0.621	–
Kim et al. [40]	–	–	1.087	–	0.657	–

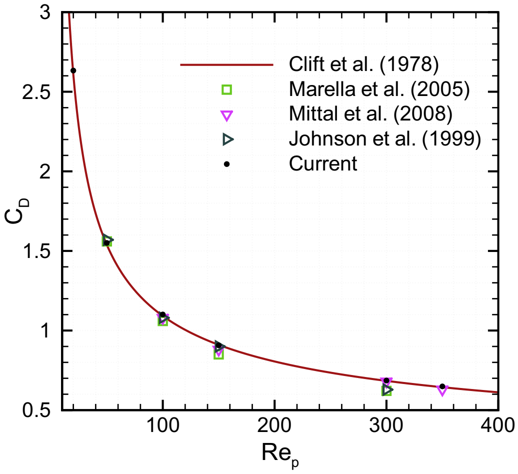
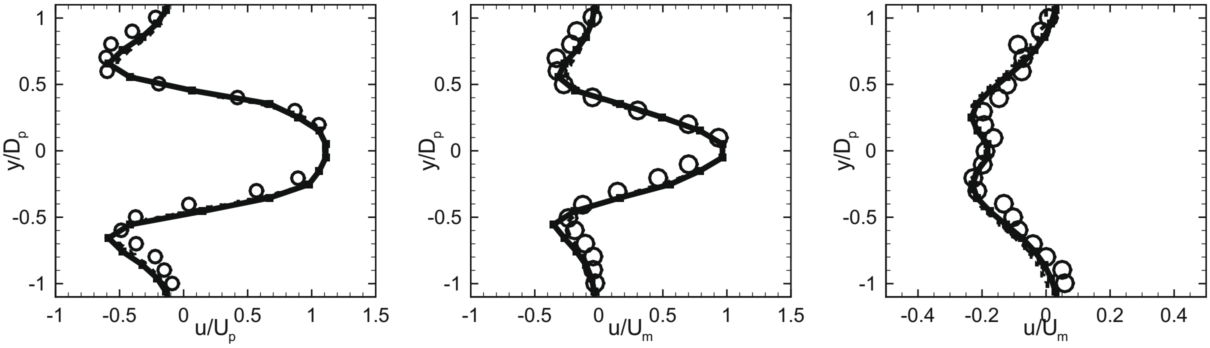


Fig. 11. Comparison of computed mean drag coefficient with experimental and numerical data: • present numerical approach, ——- [38], □ [28], ▽ [29], ▷ [36].



(a) $\omega t = 180°$ (b) $\omega t = 210°$ (c) $\omega t = 330°$

Fig. 12. Normalized axial velocity (u/U_m) variation in the y-direction at three different phase positions $(\omega t = 2\pi ft)$. The plots are shown at a fixed x-location, $x = -0.6D_P$ relative to the particle center at $\omega t = 180°$: ○ experimental data of Dütsch et al. [41], —— present numerical approach, - - - - numerical result of Kim and Choi [12].

by the oscillatory cylinder is symmetric about the horizontal symmetry axis. The vortex formation is characterized by two counter-rotating vortices. The vortical structure agrees very well those shown by Dütsch et al. [41] and by Kim and Choi [12].

5.4. Freely moving rigid particles

Below we consider a number of cases for freely moving rigid particles in laminar and turbulent flows.

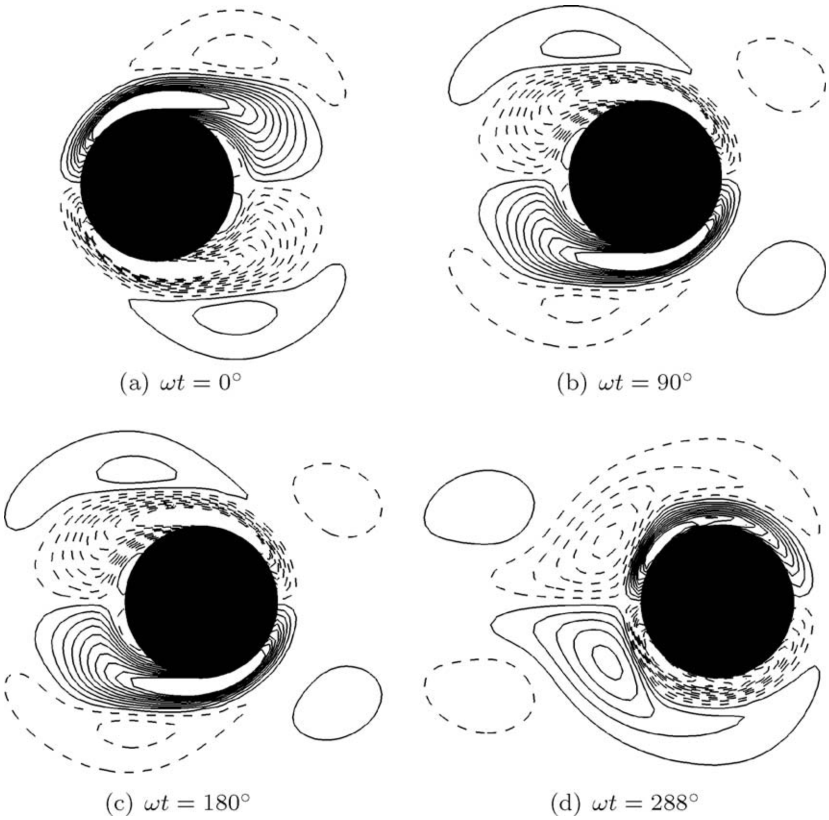


Fig. 13. Temporal evolution of spanwise vorticity in the symmetry plane at different phase angles. The normalized vorticity contours ($\omega_z D_p / U_m$) are plotted in the increments of 0.85 and range between −8.5 to 8.5. Dashed lines are negative contours.

5.4.1. The falling sphere problem

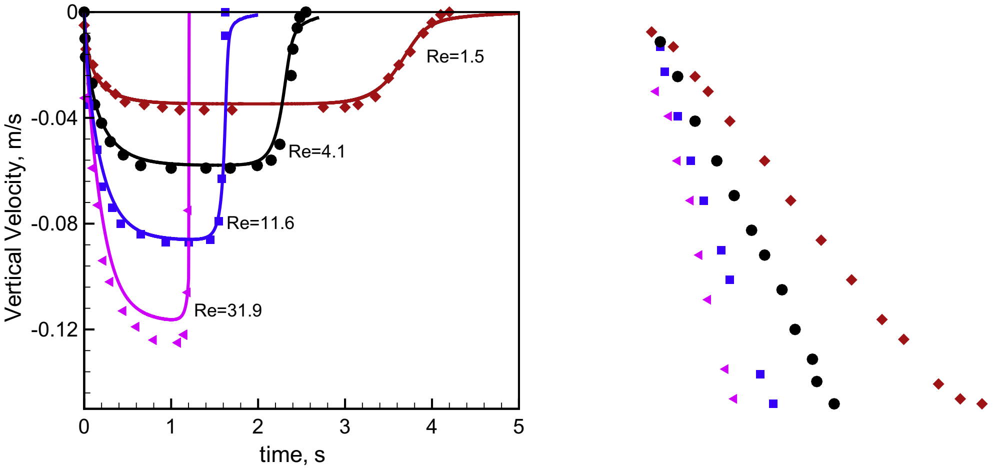
We consider the problem of a single sphere falling under gravity in a closed container. The particle density is ($\rho_p = 1120\ \mathrm{kg/m^3}$) and the diameter is ($d_p = 15$ mm). The sphere is settling in a box of dimensions $10 \times 10 \times 16\ \mathrm{cm^3}$. The particle is released at a height $H = 12$ cm from the bottom of the box. The boundaries of the box are treated as no-slip walls. The fluid properties are varied to obtain different Reynolds numbers based on the terminal velocity of the particle. The simulation conditions correspond to the experimental study by ten Cate et al. [42]. Table 5 provides detailed information about the parameters used in this test problem.

We simulate the above cases on a fine uniform grid of $100 \times 100 \times 160$ points with a grid resolution of $\Delta = 1$ mm. This provides around 15 grid points inside the particle domain. The material volumes are cubical with $\frac{\Delta}{\Delta_M} = 5$, where Δ_M is the size of the material volume. Accordingly, there are around 75 material volumes along the diameter of the spherical particle in each direction. A uniform time-step ($\Delta t = 0.5$ ms) is used for all cases. This time step is in the same range as the one used in Lattice Boltzmann simulations by ten Cate et al. [42] and simulations by Feng and Michaelides [8] based on Proteus. For this time step the CFL $\leqslant 0.1$ at all times. Later we conduct convergence study of this case with varying grid sizes and time-steps to show their effects on the solution.

Fig. 14(a) and (b) shows the comparison of the time evolution of particle settling velocity and position at different times obtained from the numerical simulations with the experimental data [42]. The simulation predictions for both the particle velocity and the particle position show good agreement with the experimental data. The slowing of the particle towards the

Table 5
Parameters for the sedimenting sphere test problem.

Case name	ρ_F ($\mathrm{kg/m^3}$)	μ_F ($10^{-3}\ \mathrm{Ns/m^2}$)	u_∞ (m/s)	$Re_p = \frac{\rho_F u_\infty d_p}{\mu_F}$
C1	970	373	0.038	1.5
C2	965	212	0.06	4.1
C3	962	113	0.091	11.6
C4	960	58	0.128	31.9



(a) Fall Velocity

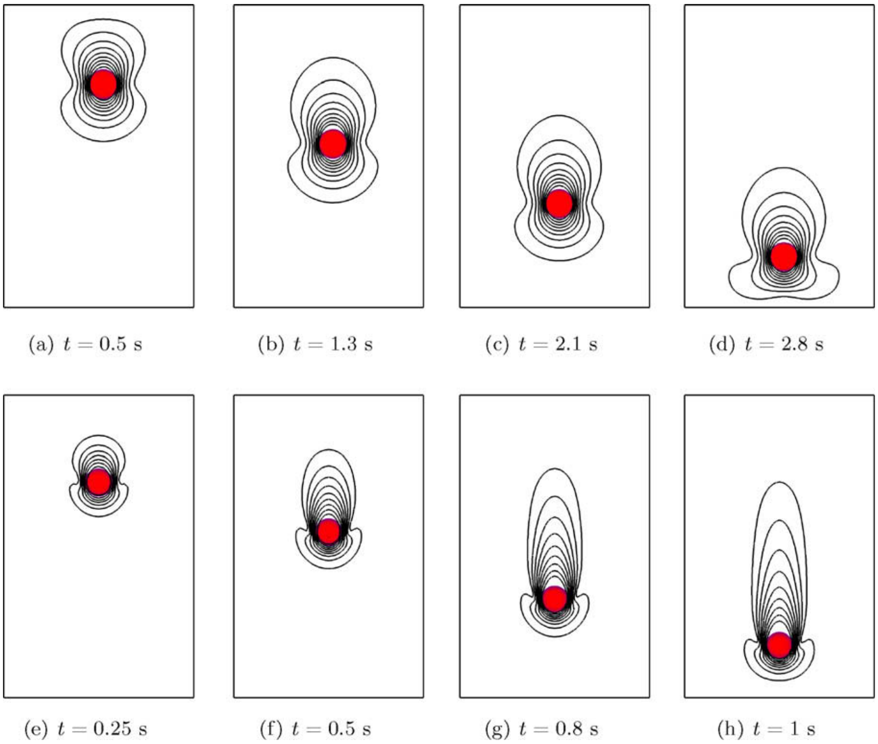
end of the simulation are to due to the presence of the bottom wall. Variations in the predicted and experimental data towards the last stages of particle settling are strongly affected by the collision model used.

It should be noted that the Lattice Boltzmann simulations conducted by ten Cate et al. [42] used a calibration procedure which computes an effective sphere radius from an analytic expression for the drag force at low Reynolds numbers at a given volumetrically averaged fluid velocity. Without this procedure, the velocity of the particle can be up to 20% different from the experimental values [42]. In the present simulations, such calibration or parameter fitting is not attempted. Use of large number of material volumes within the sphere (larger ratio between the background grid and the material volume grid) provides an accurate description of the sphere size and surface.

Fig. 15 shows the temporal evolution of the normalized velocity magnitude in the symmetry plane for $Re = 1.5$ and $Re = 31.9$. At $Re = 1.5$, the fluid velocity induced by the motion of the particle is not symmetric in the direction of motion. The evolution of fluid velocity induced at higher Reynolds number shows elongated, but narrower wake region. At both Reynolds numbers, the effect of the bottom wall is evident when the particle approaches the bottom wall. The fluid velocity contours in front of the falling particle get modified. The velocity contours compare well with those measured by ten Cate et al. [42].

Convergence study: We performed a detailed grid and time-step refinement study to investigate the order of accuracy of our numerical scheme. Accordingly, we simultaneously refined the background grid and the time-step by keeping the ratio $\frac{\Delta t}{\Delta}$ the same to investigate the collective spatial and temporal errors. We present the results for high Reynolds number case ($Re = 31.9$). Uniform Cartesian grids with five grid resolutions of $50 \times 50 \times 80$, $60 \times 60 \times 96$, $70 \times 70 \times 112$, $90\times 90 \times 144$, and $100 \times 100 \times 160$ were considered with appropriate refinements of the time-steps. The corresponding time steps used were $\Delta t = 10$, 8.33, 7.1, 5.55, and 5 ms, respectively. Note that we simultaneously vary the grid size and time-step in order to evaluate the collective temporal and spatial discretization errors. The material volume resolution was also varied with subsequent background grid refinements by keeping the ratio $\frac{\Delta}{\Delta_M} = 5$ fixed. The simulations were performed till $t = 0.8$ s and compared with fine grid solution ($120 \times 120 \times 192$) to obtain the L_∞ error. Fig. 16(a) shows the behavior of numerical error under the grid and time-step refinement. The convergence is slightly less than second-order accuracy, however, it is much improved compared to the first order accurate scheme presented by Sharma and Patankar [14] for the falling sphere problem. Fig. 16(b) shows the corresponding evolution of the particle velocity at different resolutions compared with the experimental data.

Finally, we investigate the effect of time-step refinement on the particle position and velocity. Accordingly, we use a fine grid of $100 \times 100 \times 160$ with $\frac{\Delta}{\Delta_M} = 5$ and compute the particle fall velocity at for cases C1–C4 with time-steps $\Delta t = 0.5$ and 5 ms for each. Fig. 17(a) and (b) shows the time-evolution of predicted particle position and velocity, respectively. For low Re, accurate computation of the particle motion is necessary in the particle acceleration and deceleration phases. For large time-steps, it is observed that the particle velocity is consistently over-predicted and thus the particle position is always lower than that observed in experiments. With refinement in time-steps; however, good comparisons are obtained. For larger Reynolds numbers, time-step refinement produced little change in the particle position. Recently, Feng and Michaelides [8] used their direct forcing method to simulate the sedimenting sphere problem with time steps on the order of 0.5 ms and showed similar agreement. For larger time-steps Veeramani et al. [15] show similar results as presented here. The present numerical scheme is stable and gives accurate results for particle fall velocity at CFL numbers on the order of unity. For the cases stud-


(a) $t = 0.5$ s (b) $t = 1.3$ s (c) $t = 2.1$ s (d) $t = 2.8$ s

(e) $t = 0.25$ s (f) $t = 0.5$ s (g) $t = 0.8$ s (h) $t = 1$ s

Fig. 15. Contours of normalized velocity magnitude $\left(\frac{\|u\|}{u_\infty}\right)$ at different times during the free fall for $Re = 1.5$ (a–d) and 31.9 (e–h). The time instances are chosen such that the particles are at the same height from the bottom wall for each Reynolds number. Contour lines are between 0 and 1 with equal spacing of 0.1.

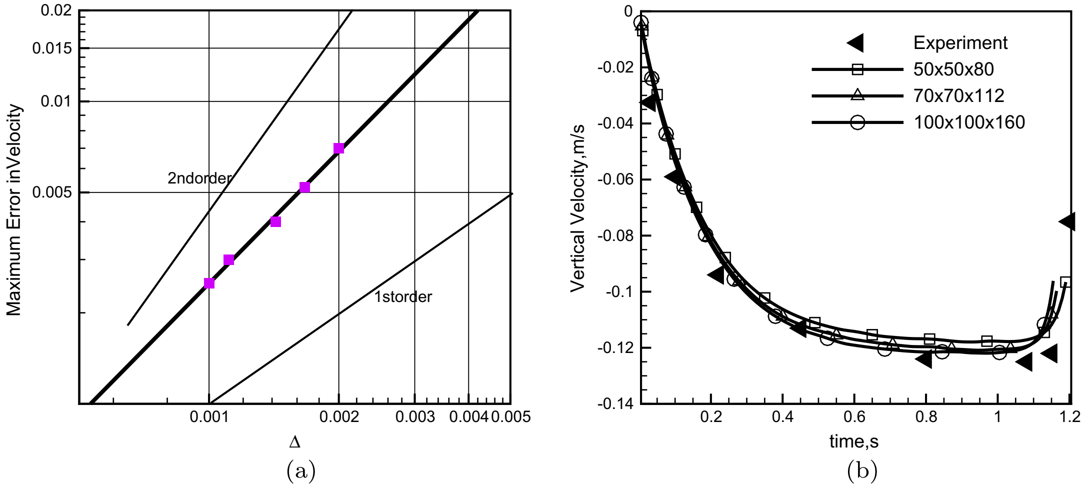
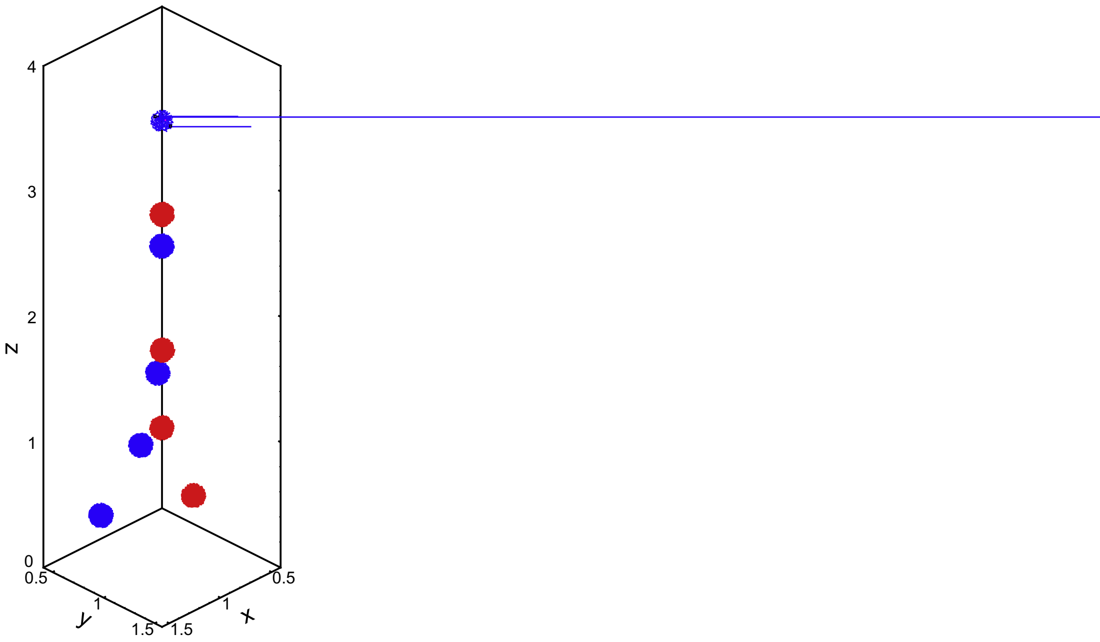


Fig. 16. Convergence studies on particle settling velocity for $Re = 31.9$ under simultaneous grid and time-step refinement: (a) L_∞ error in fall velocity, (b) comparison of fall velocity at different resolutions with experimental data.

ied here, it is apparent that the implicit formulation for computing rigid body motion and rigidity constraint force does not restrict the time steps to small values.

5.5. Wake interactions of two particles

Interactions between wakes of two particles falling under gravity are investigated to further validate our method. Two cases are considered: (i) identical particles falling under gravity and released with a certain separation distance, and (ii) same



moves to the same side and unsteady wake interactions are observed showing the capability of the numerical solver to capture such phenomena.

5.7. *Particle laden isotropic turbulent flow*

The numerical scheme is used to simulate particle-laden homogeneous, isotropic turbulent flow in a periodic box of length π with grid resolution of 128^3. A stationary isotropic turbulent flow is first developed using the linear forcing proportional to the local velocity [43]. The turbulence parameters for a Reynolds number of 54 based on the Taylor microscale. The turbulence intensity is $U' = 0.84$, the dissipation rate $\epsilon = 0.2$, the fluid density $\rho = 1$, and the kinematic viscosity $\nu = 0.013$. This gives the Kolmogorov length scale of $\eta = 0.056$, the Kolmogorov time scale is $\tau_K = 0.25$ the Taylor microscale $\lambda = 0.81$, the integral length scale of $L = 1.65$, the integral time scale $T = 1.98$, and $k_{\max}\eta$ (the measure of resolution) is 2.28. The time-step used is $\Delta t = 1 \times 10^{-3}$.

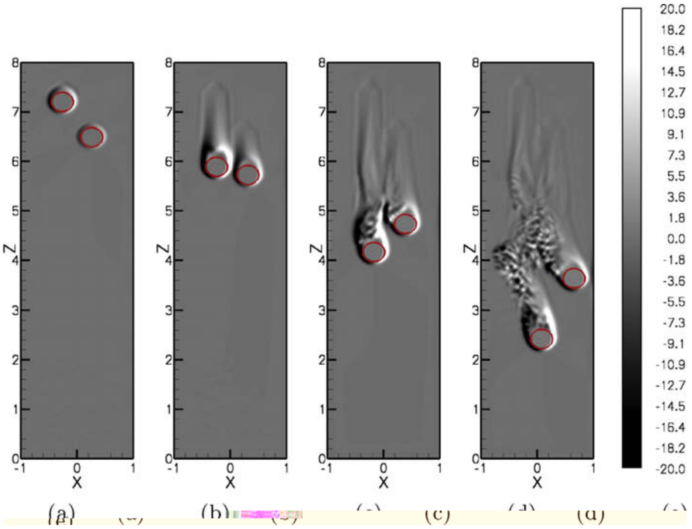


Fig. 20. Time evolution of vorticity contours showing unsteady wake interactions between two particles of same size but different densities released off-centered and falling under gravity: (a) 25 ms, (b) 75 ms, (c) 125 ms, (d) 175 ms, and (e) contour legend. Gravity is in the negative Z direction.

After obtaining a stationary state, the forcing function is turned off and 125 solid spherical particles are injected into the domain, with initial uniform distribution. The diameter of the particles (D_p) is 0.2 providing around 8 grid points over the particle domain. The particles are arranged such that they have a separation distance of $\frac{\pi}{5}$ between nearest neighbors (i.e. north, south, east, west, top and bottom). The material volume resolution is based on the ratio $\frac{\Delta}{\Delta_M} = 4$. The particle density is $\rho_p = 9$ and the particle relaxation time is $\tau_p = \frac{1}{18}\frac{\rho_p}{\rho}\frac{D_p^2}{\nu} \approx 1.77$. For the present case, the size of the particle is larger than the Kolmogorov length scale. Accordingly, different time scales can be used to normalize the particle relaxation time and define the Stokes number. Based on the Kolmogorov time-scale, the Stokes number, $St = \tau_p/\tau_K = 4.42$. The particle Reynolds number ($Re_p = \rho d_p|\mathbf{u}_{\rm rel}|/\mu$), where $\mathbf{u}_{\rm rel}$ is the relative velocity between the fluid and the particle, is on the order of 10–20 in these simulations. The relative velocity $\mathbf{u}_{\rm rel}$ is estimated when the particles are first injected into the domain and represents the characteristic velocity scale based on the fluctuating slip velocity. Also in these cases gravity is absent, and the inter-particle collision parameters are estimated based on the lubrication model used by ten Cate et al. [17]. As was shown earlier for the flow over a fixed spherical particle, the resolution of 8–12 grid cells inside the particle domain is sufficient to resolve the fluid–particle interactions in this regime.

The particles are initially at rest. As the turbulent flow decays, the particles first accelerate, reach a maximum velocity and then decelerate. Fig. 21 shows the time-evolution of the out-of-plane vorticity contours together with the location of the particles in the symmetry plane $z = 0$. Note that since a planar cut of instantaneous particle positions is shown, the particle boundaries are circles of different sizes depending upon the instantaneous location of the particle centers. Accordingly, the shapes and sizes of the circles shown in the figures appear different. However, this is just the plotting artifact and in the simulations the particles retain their size. The particles do show some clustering as they evolve from uniform distribution. The simulation was performed on 64-processors IBM machines at San Diego Supercomputing Center. It requires approximately 6 s per time-step for this simulation. The overhead of the computing time due to computation of the rigidity constraint and the motion of particles is about 10% and another 10% time is spent on inter-particle collision scheme.

To demonstrate that out approach can simulate arbitrary shaped particles, freely moving cubical particles of size 0.2 are computed in an isotropic turbulent flow. Other flow and particle parameters are same as above. Fig. 22 shows the snapshots at different times of the cubical particles and vorticity contours in the symmetry plane, again showing interactions between the particles and the unsteady turbulent motion. The initially uniformly spaced particles, cluster in small vorticity regions. Rotation of the particles around their centroids is visible.

5.8. Simulating large number of particles in turbulent flows

The numerical scheme developed is parallelized using the MPI-paradigm. Fully resolved simulations of large number of particles in turbulent flows requires complete resolution of the smallest temporal and spatial scales. For the above case of isotropic turbulent flow, the CPU-time distribution in each task of the numerical algorithm is shown in Fig. 23. For this particular case, the number of particles are roughly evenly distributed on each processor. The time required for computation of the rigidity constraint and the particle motion itself is significantly less compared to the time required for a single time-step in the absence of any particles. For this case, the collision frequency among the 125 particles was small. Inter-particle collision frequency and computation of collision force, however, may require much more CPU time than that indicated in Fig. 23

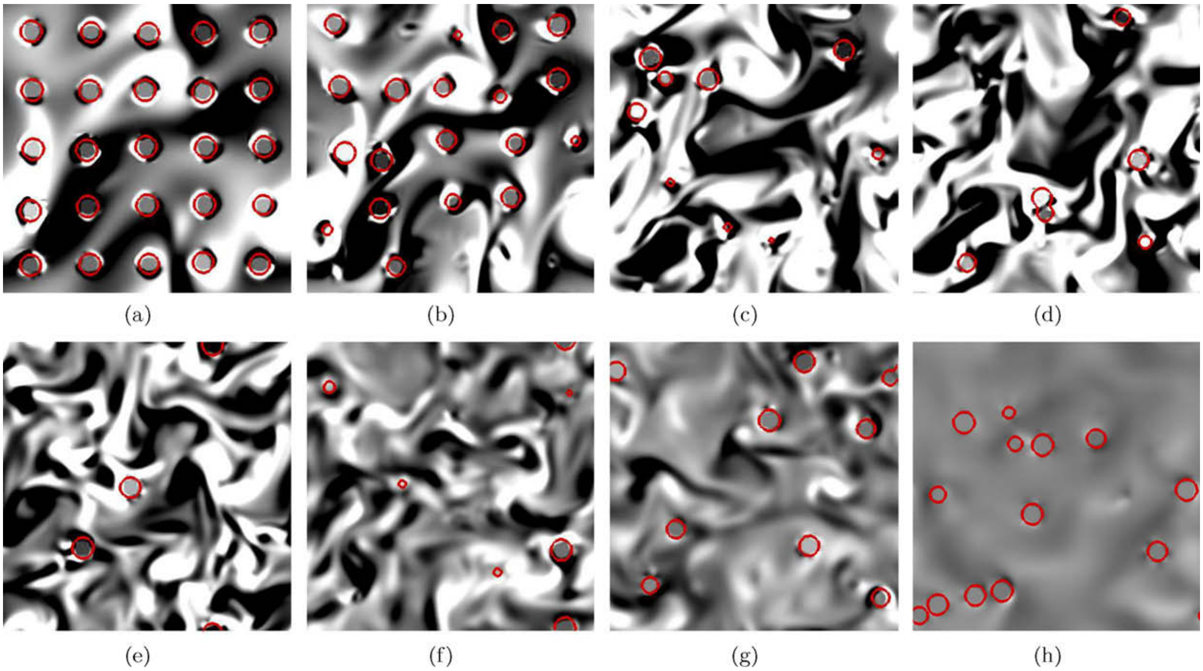


Fig. 21. Temporal evolution of spherical particles and out-of-plane vorticity in a decaying isotropic turbulence flow. The snapshots are at different times after the particles are injected into the domain, t/T: (a) 0.5, (b) 2.5, (c) 5, (d) 7.5, (e) 12.5, (f) 17.5 and (g) 20. The particle kinetic energy initially increases and then decreases. Finally, the fluid–particle system comes to rest.

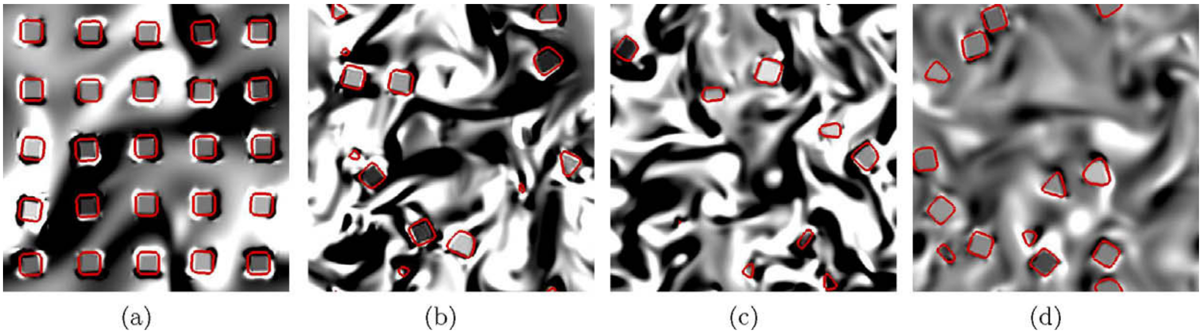


Fig. 22. Temporal evolution of cubical particles and out-of-plane vorticity in a decaying isotropic turbulence flow. The snapshots are at different times after the particles are injected into the domain, t/T: (a) 0.5, (b) 5, (c) 10, and (f) 15.

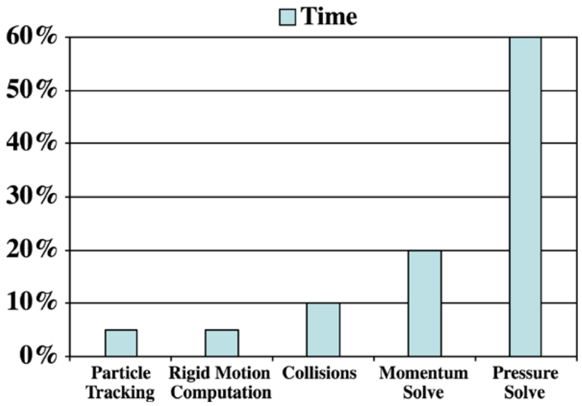


Fig. 23. Computational time distribution in each time-step for rigid-particle laden isotropic turbulent flow.

owing to the N^2-nature of the collision computation. Methods based on Verlet lists and linked-lists [23] can be used to reduce this time. The overall solution is then constrained by the solution of the Poisson system, indicating that the computational overhead of simulating particle motion is small.

Simulation of large number of particles on the order of 10,000 particles and beyond in complex turbulent flows is feasible with the numerical scheme. However, several issues related to numerical efficiency need to be addressed. For example, if the particles are not uniformly distributed over the computational domain, and the grid partitioning is based on the fluid flow solver, severe load imbalance may occur. Advanced strategies for domain decomposition are required and will be the focus of subsequent study.

6. Summary and conclusions

A numerical formulation for fully resolved simulations of freely moving rigid particles in laminar and turbulent flows is developed based on a co-located grid, finite-volume method. In this fictitious-domain based approach, the entire computational domain is first treated as a fluid of density corresponding to the fluid or particle densities in their respective regions. The incompressibility and rigidity constraints are applied to the fluid and particle regions, respectively, by using a fractional step algorithm. The approach extends the formulation developed by Patankar [1] to time-staggered, co-located grids. Use of consistent interpolations between the particle material volumes and the background grid and parallel implementation of the algorithm facilitates accurate and efficient simulations of large number of particles. Implementation of this approach in finite-volume, co-located grid based numerical solver is presented. The numerical approach is validated for flow over a fixed sphere at various Reynolds numbers and flow generated by a periodically oscillating cylinder. Several fluid–structure interaction problems were also studied to investigate the accuracy and predictive capability of the scheme for freely moving particles. Finally, simulation spherical particles in a decaying isotropic turbulent flow is performed to show the feasibility of simulations of turbulent flows. The overhead due to presence of particles and computing their motion is small and the computational speed is governed by the pressure Poisson equation used to impose the incompressibility constraint, making this approach attractive for large-scale simulations resolving the multiscale interactions between particles and turbulent flow.

Acknowledgments

SVA and MM acknowledge Dr. Ki-Han Kim of the Office of Naval Research for the support provided under the ONR Grant No. N000140610697, which resulted in successful completion of the numerical implementation of the algorithm. The computing time at San Diego Supercomputing Center's Datastar machine is highly appreciated. We are also thankful to Prof. Haecheon Choi of Seoul National University for providing the dataset for comparison on flow induced by a periodically oscillating circular cylinder. NAP acknowledges support from NSF CAREER Grant CTS-0134546. All simulations were performed in SVA's group at Oregon State University.

References

[1] N. Patankar, A formulation for fast computations of rigid particulate flows, Center for Turbulence Research Annual Research Briefs 2001 (2001) 185–196.
[2] H. Hu, N. Patankar, M. Zhu, Direct numerical simulations of fluid–solid systems using the arbitrary Lagrangian–Eulerian technique, Journal of Computational Physics 169 (2) (2001) 427–462.
[3] A. Johnson, T. Tezduyar, Simulation of multiple spheres falling in a liquid-filled tube, Computer Methods in Applied Mechanics and Engineering 134 (3–4) (1996) 351–373.
[4] R. Glowinski, T. Pan, T. Hesla, D. Joseph, 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 169 (2) (2001) 363–426.
[5] C. Peskin, The immersed boundary method, Acta Numerica 11 (2003) 479–517.
[6] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review of Fluid Mechanics 37 (1) (2005) 239–261.
[7] A. Ladd, R. Verberg, Lattice-Boltzmann simulations of particle–fluid suspensions, Journal of Statistical Physics 104 (2001) 1191–1251.
[8] Z. Feng, E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, Journal of Computational Physics 202 (1) (2005) 20–51.
[9] Z. Zhang, A. Prosperetti, A second-order method for three-dimensional particle simulation, Journal of Computational Physics 210 (1) (2005) 292–324.
[10] K. Taira, T. Colonius, The immersed boundary method: a projection approach, Journal of Computational Physics 225 (2) (2007) 2118–2137.
[11] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, Journal of Computational Physics 209 (2) (2005) 448–476.
[12] D. Kim, H. Choi, Immersed boundary method for flow around an arbitrarily moving body, Journal of Computational Physics 212 (2) (2006) 662–680.
[13] N. Patankar, P. Singh, D. Joseph, R. Glowinski, T. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, International Journal of Multiphase Flow 26 (9) (2000) 1509–1524.
[14] N. Sharma, N. Patankar, A fast computation technique for the direct numerical simulation of rigid particulate flows, Journal of Computational Physics 205 (2) (2005) 439–457.
[15] C. Veeramani, P. Minev, K. Nandakumar, A fictitious domain formulation for flows with rigid particles: a non-Lagrange multiplier version, Journal of Computational Physics 224 (2) (2007) 867–879.
[16] T. Kajishima, S. Takiguchi, Y. Miyake, Modulation and subgrid scale modeling of gas-particle turbulent flow, in: Recent Advances in DNS and LES: Proceedings of the Second Afosr Conference Held at Rutgers, the State University of New Jersey, New Brunswick, USA, June 7–9, 1999.
[17] A. Cate, J. Derksen, L. Portella, H. van den Akker, Fully resolved simulations of colliding monodisperse spheres in forced isotropic turbulence, Journal of Fluid Mechanics 519 (2004) 233–271.
[18] K. Mahesh, G. Constantinescu, P. Moin, A numerical method for large-eddy simulation in complex geometries, Journal of Computational Physics 197 (1) (2004) 215–240.
[19] K. Mahesh, G. Constantinescu, S. Apte, G. Iaccarino, F. Ham, P. Moin, Large-eddy simulation of reacting turbulent flows in complex geometries, Journal of Applied Mechanics 73 (2006) 374.

[20] P. Koumoutsakos, Multiscale flow simulations using particles, Annual Review of Fluid Mechanics 37 (1) (2005) 457–487.
[21] A. Roma, C. Peskin, M. Berger, An adaptive version of the immersed boundary method, Journal of Computational Physics 153 (2) (1999) 509–534.
[22] B. Maury, A many-body lubrication model, Comptes Rendus de l'Academie des Sciences Series I Mathematics 325 (9) (1997) 1053–1058.
[23] L. Verlet, Computer experiments on classical fluids. I. Thermodynamical properties of Lennard–Jones molecules, Physical Review 159 (1) (1967) 98–103.
[24] R. Hockney, J. Eastwood, Computer Simulation Using Particles, Institute of Physics Publishing., 1988.
[25] D. Kim, H. Choi, A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids, Journal of Computational Physics 162 (2) (2000) 411–428.
[26] C. Pierce, P. Moin, Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion, Journal of Fluid Mechanics 504 (2004) 73–97.
[27] Lawrence Livermore National Laboratory's CASC Team, Scalable linear solvers, HYPRE, hypre2.0.0 360. <http://www.llnl.gov/CASC/linear_solvers>.
[28] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface cartesian grid method I: an easily implemented technique for 3d moving boundary computations, Journal of Computational Physics 210 (1) (2005) 1–31.
[29] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, Journal of Computational Physics 227 (10) (2008) 4825–4852.
[30] C. Williamson, The natural and forced formation of spot-like'vortex dislocations' in the transition of a wake, Journal of Fluid Mechanics 243 (1992) 393–441.
[31] Y. Zang, R. Street, J. Koseff, A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates, Journal of Computational Physics 114 (1) (1994) 18–33.
[32] R. Henderson, Details of the drag curve near the onset of vortex shedding, Physics of Fluids 7 (1995) 2102.
[33] R. Mittal, S. Balachandar, On the inclusion of three-dimensional effects in simulations of two-dimensional bluff-body wake flows, in: Proceedings, 1997 ASME Fluids Engineering Division Summer Meeting.
[34] C. Shu, N. Liu, Y. Chew, A novel immersed boundary velocity correction–lattice Boltzmann method and its application to simulate flow past a circular cylinder, Journal of Computational Physics 226 (2) (2007) 1607–1622.
[35] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries, Journal of Computational Physics 156 (2) (1999) 209–240.
[36] T. Johnson, V. Patel, Flow past a sphere up to a Reynolds number of 300, Journal of Fluid Mechanics 378 (1999) 19–70.
[37] S. Taneda, Experimental investigation of the wake behind a sphere at low Reynolds numbers, Journal of the Physical Society of Japan 11 (10) (1956) 1104–1108.
[38] R. Clift, J. Grace, M. Weber, Bubbles, Drops, and Particles, Academic Press, New York, 1978.
[39] R. Mittal, A Fourier–Chebyshev spectral collocation method for simulating flow past spheres and spheroids, International Journal for Numerical Methods in Fluids 30 (7) (1999) 921–937.
[40] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, Journal of Computational Physics 171 (1) (2001) 132–150.
[41] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers, Journal of Fluid Mechanics 360 (1998) 249–271.
[42] A. ten Cate, C. Nieuwstad, J. Derksen, H. van den Akker, Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity, Physics of Fluids 14 (2002) 4012.
[43] T. Lundgren, Linearly forced isotropic turbulence, Center for Turbulence Research Annual Research Briefs 2003 (2003) 461–473.