



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Mesoscale simulations of particulate flows with parallel distributed Lagrange multiplier technique

Yuliya Kanarska

May 10, 2010

ICMF 2010
Tampa, FL, United States
May 30, 2010 through June 4, 2010

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Mesoscale simulations of particulate flows with parallel distributed Lagrange multiplier technique

Y. Kanarska*

* Lawrence Livermore National Laboratory, Livermore, CA 94550, USA

kanarska1@llnl.gov

Keywords: particle-laden flows, fictitious domain method

Abstract

Fluid particulate flows are common phenomena in nature and industry. Modeling of such flows at micro and macro levels as well establishing relationships between these approaches are needed to understand properties of the particulate matter. We propose a computational technique based on the direct numerical simulation of the particulate flows. The numerical method is based on the distributed Lagrange multiplier technique following the ideas of Glowinski et al. (1999). Each particle is explicitly resolved on an Eulerian grid as a separate domain, using solid volume fractions. The fluid equations are solved through the entire computational domain, however, Lagrange multiplier constrains are applied inside the particle domain such that the fluid within any volume associated with a solid particle moves as an incompressible rigid body. Mutual forces for the fluid-particle interactions are internal to the system. Particles interact with the fluid via fluid dynamic equations, resulting in implicit fluid-rigid-body coupling relations that produce realistic fluid flow around the particles (i.e., no-slip boundary conditions). The particle-particle interactions are implemented using explicit force-displacement interactions for frictional inelastic particles similar to the DEM method of Cundall et al. (1979) with some modifications using a volume of an overlapping region as an input to the contact forces. The method is flexible enough to handle arbitrary particle shapes and size distributions. A parallel implementation of the method is based on the SAMRAI (Structured Adaptive Mesh Refinement Application Infrastructure) library, which allows handling of large amounts of rigid particles and enables local grid refinement. Accuracy and convergence of the presented method has been tested against known solutions for a falling sphere as well as by examining fluid flows through stationary particle beds (periodic and cubic packing). To evaluate code performance and validate particle contact physics algorithm, we performed simulations of a representative experiment conducted at the University of California at Berkley for pebble flow through a narrow opening.

Introduction

Particulate flows occur in a wide range of industrial applications and in nature. Clearly, it's difficult to have one single simulation method that can cover all length and time scales. Currently there is a hierarchy of methods that can cover different length and time scales with different levels of details (Zhu et al. 2007).

When the computational grid size is much larger than particle size usually a two-fluid (or multiphase) approach is used. The computational fluid dynamics two-fluid approach is often associated with methods described in Gidaspow (1994) and various implementations are available, such as the commercial code FLUENT (FLUENT 2001) and the DOE/NFTL code MFI (Syamlal et al. 1993). These two-fluid codes usually utilize a granular-kinetic-theory based constitutive model to represent the "fluid" comprised of fluidized particles,

and empirical two-way coupling relations between fluid and particles. An advantage of the multi-fluid model is that in principle it can be used to compute any multi-phase flow regime. However the effective use of these models strongly depends on the constitutive or closure relations for the solid phase and momentum exchange between phases. For two-phase systems comprised of billions of particles (like, for example, most fluidized-beds or pneumatic-transport systems for fine particulates) such continuum models are the only computationally viable simulation methods available. In fact, development of a general theory to correctly represent granular flow with fluid as a continuum is still a challenging research area.

If the cell size is larger than the particle size, a combined CFD-DEM coupling approach is used. In this approach, the motion of individual particles is obtained by

solving Newton’s equations of motion, while the flow of continuum gas is determined by the CFD on a computational cell scale. A variety of continuum fluid codes, coupled with discrete-element-method (DEM) particles, are utilized by researchers around the world in commercial codes as well as codes developed by university researchers. An overview of these methods can be found in Zhu et al. (2007). The algorithm relies on the parameterizations of drag terms similar to the Ergun equation (Ergun 1952) for static packed beds, or the Wen-Yu equation for moving beds (Wen and Yu 1966). While some questions regarding parameterizations of drag terms are still remaining, this approach might be suitable for simulations of intermediate-scale system (about million of particles) and promises to be a very powerful tool. However, there are some restrictions of the algorithm and the CFD-DEM coupling algorithm since it assumes that the cell size in the CFD model should be larger than the particle size. This may result in using a fairly coarse mesh in the areas (nozzles, openings) where only few particles across an important geometric feature are considered as well as placing severe restrictions on the maximum particle size that can be included in simulations.

And, finally, if the cell size is smaller than the particle diameter the direct numerical simulations, resolved in the particle and fluid domains, can be applied. In this case there are no assumed drag terms. Drag effects and the flow around each particle are explicitly resolved. And particle-particle interactions are modeled, again, using a DEM-like approach. The method has no assumptions for drag terms and can be used to improve fluid-coupling terms and derive closure parameterizations that can represent the effective coarse-grained interactions in the larger scale models just mentioned above. Moreover, rapid granular flows demonstrate lack of scale separations which render local closure laws inapplicable in many applications and will require multi-scale approach if it is not feasible to use a high resolution grid everywhere.

A variety of fixed-grid and mesh-free methods have been utilized for simulation of fluid-particle systems. We have selected the Lagrange multiplier technique following the ideas of Glowinski et al. (1999) and Patankar et al. (2001) to model fluid-particle systems. The advantage of the method is that its finite element formulation permits the use of a fixed structured grid. This eliminates the need for remeshing the domain, a necessity in the unstructured grid-based methods. The objective of this paper is to present an efficient approach based on the advantages of utilization of a stationary Eulerian grid with adaptive mesh refinement (AMR). A parallel implementation of the method is based on the SAMRAI library, which allows handling of large amounts of rigid

particles and enables local grid refinement in the areas where higher resolutions are needed.

Numerical Model

The method we used is based on the Distributed Lagrange Multiplier (DLM) technique of Glowinski et al. (1999) and Patankar et al. (2001) that was originally developed to study particulate suspension flows. The code uses a stationary Eulerian grid. Particle positions are treated as Lagrangian variables. The particle domain is explicitly resolved on the Eulerian grid using solid volume of fractions. The idea of the method is to solve the fluid equation in the entire domain, and then correct the flow inside the rigid domain using Lagrange multipliers. The original works are based on elastic collision forces that prevent particles from overlap. We extended this approach by incorporating DEM methods for inelastic, frictional contact forces. The governing equations are solved using a fractional-step scheme for time discretization. The fluid equations are solved in the entire computational domain at the first stage. It results in a provisional divergence-free intermediate velocity field. At the next stage, the constraint of rigid motion (in the form of Lagrange multiplier) is applied in the solid domain. A rigid body motion is imposed by constraining the deformation-rate tensor within the particle domain to be zero. The code is parallelized using the SAMRAI framework developed at LLNL Hornung and Kohn (2002). This framework allows to track individual particle position on multiple cpus and refining the resolution on a structured grid in areas of interest (e.g. solid-fluid interfaces, maximum vorticity zones etc.).

Collisionless governing equations

The idea of the Lagrange multiplier algorithm is based on the formulations of Glowinski et al. (1999), Patankar et al. (2001) and Sharma and Patankar (2005). The particle domain is denoted as $P(t)$, where ∂P is the interface between the particle and the fluid. F is the fluid domain that is not shared with the particles. The entire computational domain that includes both the fluid and the particles is denoted by $F \cup P$. The governing equations in the fluid domain can be written as:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} + \mathbf{f}, \text{ in } F \cup P \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \text{ in } F \cup P \quad (2)$$

$$D[\mathbf{u}] = 0, \text{ in } P(t) \quad (3)$$

$$D[\mathbf{u}] \cdot \mathbf{n} = 0, \text{ on } \partial P(t) \quad (4)$$

$$\mathbf{u} = \mathbf{u}_s, \text{ in } P(t) \quad (5)$$

where \mathbf{u} is the fluid velocity, p is the pressure, μ is the fluid viscosity, ρ is the density that is equal to ρ_f in the fluid domain and equal to ρ_s in the particle domain, \mathbf{g} is a body force, and \mathbf{n} is a unit normal on the particle surface. The rigid body velocity inside the particle \mathbf{u}_s is represented as

$$\mathbf{u}_s = \mathbf{U} + \mathbf{\Omega} \times \mathbf{r}, \quad (6)$$

where \mathbf{U} and $\mathbf{\Omega}$ are the translational and angular velocities of the particle, respectively, and \mathbf{r} is the position vector of the point with respect to the particle centroid. Force \mathbf{f} that appears in the right hand side of the momentum equation is non-zero only in the particle domain and arises as a result of the rigid body motion constraint in the particle domain. $D[\mathbf{u}]$ is the deformation-rate tensor defined as

$$D[\mathbf{u}] = \frac{1}{2}[\nabla \mathbf{u} + \nabla \mathbf{u}^T] \quad (7)$$

The stress tensor is then given by:

$$\sigma = -p\mathbf{I} + \tau, \quad (8)$$

where \mathbf{I} is the identity tensor, p is the pressure and τ is the viscous stress tensor given by

$$\tau = \mu \mathbf{D}[\mathbf{u}], \quad (9)$$

for a Newtonian fluid, where μ is the viscosity of the fluid. The particle domain in this formulation is treated as a fluid with an additional constraint (Lagrange multiplier) to impose the rigid body motion in such a way that the deformation-rate tensor $D[\mathbf{u}]$ within the particle domain is zero. It should be noted that the representation (6) for solid velocity is the sufficient and necessary for condition (7) to be true. Stresses can develop in the rigid fluid domains, but the only displacements allowed are strain-free translations and rotations. The numerical algorithm to solve the fluid-particle equations of motion is presented in the next section.

The numerical algorithm

The integration of the governing equations is done using a fractional time stepping approach in the time interval $[t^n, t^{n+1}]$. In the present algorithm, the velocity and pressure are cell-centered quantities. The velocity is defined at integer multiples of Δt , whereas the pressure is defined at half-timesteps. The system of equations (1),(3) with boundary conditions (4)-(5) is solved numerically using an operator-splitting technique that

combines the incompressibility condition and advection-diffusion as a first step; and the constraint of rigid-body motion in the particle domain and the related distributed Lagrange multiplier technique as a next step.

First step: Solve Navier-Stokes equations in the entire domain. At the first step we solve the fluid equations of motion in the entire computational region and satisfy the provisional divergence-free intermediate velocity field $\tilde{\mathbf{u}}$ using an implicit pressure projection technique.

$$\rho \frac{\tilde{\mathbf{u}} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1/2} = \mathbf{R}^{n+1/2}, \quad (10)$$

$$\nabla \cdot \tilde{\mathbf{u}} = 0, \quad (11)$$

where $\mathbf{R}^{n+1/2}$ represents all terms on the right-sides of the momentum equations except the pressure gradient terms. An unsplit second-order Godunov procedure is used to approximate the nonlinear advection term that appears in the momentum equations using both velocities defined at the centers of the Cartesian grid as well as velocities defined at the cell faces. The MAC projection method of Bell et al. (1991) that corrects divergence of advection velocity along with MUSCL advection scheme of Colella et al. (1985) are used to advect the fluid. The Crank-Nicholson scheme is used to compute diffusion terms. A divergence constraint is satisfied using the approximate pressure projection method of Almgren et al. (1996). The density is computed as $\rho = \rho_s \phi + \rho_f (1 - \phi)$, where ϕ is the solid volume fraction which is equal to 1 in the solid domain and equal to 0 in the fluid domain.

Second step: Rigid body projection in the particle domain. At the next stage, the constraint of rigid body motion (in the form of Lagrange multiplier) is applied in the solid region. The particle velocity \mathbf{u}_s in a given cell is split into translational, \mathbf{U} , and rotational, $\mathbf{\Omega}$, parts as

$$\mathbf{u}_s = \mathbf{U} + \mathbf{\Omega} \times \mathbf{r}, \quad (12)$$

where \mathbf{r} is a vector which connects the particle's centroid and a center of the considered grid cell. Particle velocities \mathbf{U} and $\mathbf{\Omega}$ are computed by integration of the provisional velocity field $\tilde{\mathbf{u}}$ in the solid domain as

$$M\mathbf{U} = \int_{\mathbf{P}} \rho_s \tilde{\mathbf{u}} d\mathbf{V}, \quad (13)$$

$$\mathbf{I}_p \mathbf{\Omega} = \int_{\mathbf{P}} \mathbf{r} \times \rho_s \tilde{\mathbf{u}} d\mathbf{V}, \quad (14)$$

where M is the mass of a particle and I_p is the moment of inertia of a particle.

The velocity field \mathbf{u}^{n+1} is then updated in the solid, fluid and mixed (where both solid and fluid are pre-

sented) domains as

$$\begin{cases} \mathbf{u}^{n+1} = \mathbf{u}_s, & \text{in solid cells} \\ \mathbf{u}^{n+1} = \tilde{\mathbf{u}}, & \text{in fluid cells} \\ \mathbf{u}^{n+1} = \phi \mathbf{u}_s + (\phi - 1) \tilde{\mathbf{u}}, & \text{in mixed cells} \end{cases} \quad (15)$$

where ϕ is the solid volume of fraction, computed based on the particle position at time n . This step could be considered as finding force \mathbf{f} that modifies provisional velocity field $\tilde{\mathbf{u}}$ such that the final velocity field \mathbf{u}^{n+1} satisfies the rigid body motion constraints. Therefore the force \mathbf{f} can be defined as

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} + \frac{\mathbf{f} \Delta t}{\rho}. \quad (16)$$

Also as it was mentioned before the final velocity should provide zero deformation-rate tensor as

$$D[\mathbf{u}^{n+1}] = 0. \quad (17)$$

Since in the method considered here the final velocity in the solid domain is known at this step and it is equal to \mathbf{u}_s , condition (17) is satisfied automatically and the final velocity is calculated from equations (15) directly. This approach is similar to one implemented in Patankar et al. (2001). In other versions of the Lagrange multiplier method (Veeramani et al. 2006; Glowinski et al. 1999) the governing equations (1)-(3) include equations of motion of each individual particle to find \mathbf{U} and $\mathbf{\Omega}$. Therefore the resulting equations are solved iteratively.

Third step: Apply particle collision forces. If the particle Reynolds number and solid volume fraction are low, particles do not interact and \mathbf{U}^{n+1} and $\mathbf{\Omega}^{n+1}$ defined at the previous step are the final velocities that describe the velocity field inside the particle. When the concentration of particles is high enough they begin to interact with each other. If the fluid viscosity is high it would prevent particles from colliding with each other. But if fluid forces are insufficient to prevent particle contacts, the separately applied collision forces \mathbf{F}^{n+1} similar to DEM models (Cundall et al. 1979) are used in our code. Our formulation is based on a visco-elastic soft collision model. Since we resolve the particle domain on a stationary Eulerian grid, we compute collision force in each cell of the overlapping region and then integrate them over the entire overlapping volume. This is the main difference with the classical DEM approach Cundall et al. (1979). We will talk more about the calculation of collisional force $\mathbf{F}^{n+1/2}$ in a separate section. Once this force is calculated, the final particle velocities (translational and rotational) are updated using the velocity field $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{\Omega}}$ from the previous time step as:

$$\mathbf{U}^{n+1} = \tilde{\mathbf{U}}^{n+1} + \mathbf{F}^{n+1/2} \Delta t, \quad (18)$$

$$\mathbf{\Omega}^{n+1} = \tilde{\mathbf{\Omega}}^{n+1} + [\mathbf{r} \times \mathbf{F}^{n+1/2}] \Delta t, \quad (19)$$

where $\tilde{\mathbf{U}}^{n+1}$ and $\tilde{\mathbf{\Omega}}^{n+1}$ are particle velocities computed at the previous stage.

Fourth step: Update particle position: Explicitly update particle position \mathbf{X}^{n+1} by the following procedure:

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2} \Delta t, \quad (20)$$

where \mathbf{X} is the position of the particle centroid. If a particle or a solid body is represented as a polyhedron with vertex coordinates \mathbf{X}_v^{n+1} , their positions are updated as

$$\mathbf{X}_v^{n+1} = \mathbf{X}^n + [\mathbf{\Omega}^{n+1} \times \mathbf{r}], \quad (21)$$

Once the vertex positions are defined the volume fractions for solid bodies are computed and used for computations at the next time step.

Treatment of collisions

Different collision models have been developed for the coupled solvers for the fluid and solid systems. These collision models aim to capture the collision process of solid particles by introducing short-range forces as additional body forces acting on the particles. Treatment of collisions includes a contact detection algorithm and applying a collision forces to prevent particles from overlapping. In the present work the contact is detected in the region where two particles overlap based on the volume of fraction function ϕ . The condition $\phi > 1$ means that more than one particle exists in the given grid cell, therefore additional collision force is applied in each grid cell where particle overlap. While this approach would not be optimal for normal DEM algorithms, it is a natural fit for fluid-particle problems. Moreover, it easily extends to non-spherical objects which are hard to accommodate in a typical DEM approach. The collision force acts along the normal (\mathbf{F}_n -component) as well as the tangential direction (\mathbf{F}_t -component) at the point of contact between two particles. For spherical particles, the normal and tangential directions are defined by the line joining the centers of two colliding particles and two lines perpendicular to it, respectively. The components of the normal collisional force are computed in each cell where particles overlap and then integrated over the overlapped region as

$$\mathbf{F}_n = - \sum_{ijk} (k_n V_{ijk} + d_n V_{ijk} v_n) / m_p, \quad (22)$$

where V_{ijk} is the volume of cell ijk in the overlapped region, v_n is the relative velocity between the two interacting particles in the normal direction, k_n is the normal

spring constant (or stiffness), d_n is the damping coefficient in the normal direction, and m_p is the particle mass.

The tangential component is computed according to a Coulomb friction law as

$$F_t = \begin{cases} \sum_{ijk} d_t V_{ijk} v_t & , |F_t| \leq \mu_f |F_n| \\ -\mu_f |F_n| t, & |F_t| > \mu_f |F_n| \end{cases} \quad (23)$$

where μ_f is the friction coefficient, and d_t is the damping coefficient in the tangential direction.

An important advantage of the present collision method is that it is flexible enough to handle arbitrary particle shapes and size distributions and doesn't require extra parametrizations for fluid-particle interactions. The shear friction forces discussed above can allow only slow movement in the tangential direction, but cannot stop or reverse tangential motion. This formulation is inadequate for applications that require truly static friction, such as heap formation or angles of repose. In such situations, there is a threshold force below which the grains do not move at all, opposed by static friction. However, implementation of even a simple history-dependent threshold rule is algorithmically complicated and will be a subject of future work. Thus the applications in this paper are limited by the viscoelastic collisional model described above.

According to the Hertzian contact theory Hertz (1882), the relation between the normal force F_n and displacement δ is given by

$$F_n = K_H \delta^{3/2}, \quad (24)$$

where K_H is the Hertzian stiffness. In the case of two spheres of the same size with radius r , K_H is expressed by

$$K_H = \frac{\sqrt{2r}E}{3(1-\sigma^2)}, \quad (25)$$

where E is Young's modulus and σ is the Poisson ratio of the particles. Since in (22) the normal contact force is proportional to the volume of the overlapping region, our computational stiffness can be related to the Hertzian stiffness as

$$K_H = k_n \pi r \delta^{1/2} \quad (26)$$

Parallelization

The code has been built on top of the SAMRAI (Structured Adaptive Mesh Refinement) library developed at LLNL Hornung and Kohn (2002). SAMRAI is a general object-oriented software infrastructure for implementing parallel scientific applications that employ structured

adaptive mesh refinement. The method uses a hierarchical structured grid approach first developed by Berger and Olinger (1984). In particular, AMR is based on a sequence of nested grids with successively finer spacing in both time and space. Increasingly finer grids are recursively embedded in coarse grids until the solution is sufficiently resolved. An error estimation procedure evaluates where additional refinement is needed and grid generation procedures dynamically create or remove rectangular fine grid patches as resolution requirements change. Automatic regridding in time is based on Richardson extrapolation and in space on detection of gradients (velocity, scalar etc) in the solution. SAMRAI provides the backbone of our implementation, managing the locally-refined Cartesian grid patch hierarchy with both the Eulerian and Lagrangian data points defined on the hierarchy. It also provides facilities for performing adaptive regridding, load balancing, and parallel data communication. To store and manage the Lagrangian data points a version of the SAMRAI IndexData patch data type is used. For a general-purpose solver library, we have chosen PETSc (Balay et al. 2004), developed at Argonne National Laboratory. This suite solves large-scale linear and nonlinear equations. We used preconditioned Krylov methods provided by this library. A parallel data managing and implementation is done similar to the algorithm described in Griffith et al. (2001).

Validation against empirical data and experiments

Fixed particle beds

Flow behavior through packed beds of spheres or other porous-media-like structures are of crucial importance in industry and nature. The determination of pressure drop through a packed bed as a function of fluid flow rate, geometrical constraints of the bed and physical properties of bed material is very critical, for example, in hydraulic and pneumatic devices. The well-known empirically derived equation used for that purpose has been proposed by Ergun (Ergun 1952) based on experimental measurements:

$$\frac{\Delta p}{L} = A \frac{(1-\varepsilon)^2}{\varepsilon^3} \frac{\nu}{D_p^2} u + B \frac{(1-\varepsilon)}{\varepsilon^3} \frac{\rho_f}{D_p} u^2, \quad (27)$$

where Δp is the pressure drop through the packed bed, L is the bed length or height, $\varepsilon = \frac{V_f}{V}$ is the bed porosity, where V_f is the voids volume, V is the total volume, u is the superficial velocity at the exit of bed, ν is the fluid dynamic viscosity, and D_p is the particle diameter. Ergun equation (27) incorporates momentum losses due to viscous effects (first term in (27) which is important in the laminar regime) and inertia effects (the second term

in (27) which dominates in the turbulent regime). The Ergun equation is often used in a more general form by introducing a friction coefficient Λ which is defined as

$$\Lambda = A + BRe^C = \frac{\Delta p}{L} \frac{\epsilon^3 D_p^2}{(1 - \epsilon)^2 \nu u}, \quad (28)$$

where Reynolds number is defined as $Re = \frac{\rho_f D_p u^2}{\nu(1 - \epsilon)}$.

A standard form of the Ergun equation uses the values of empirical constants $A = 150$, $B = 1.75$, $C = 1$. However many other studies have been performed and published to check the appropriate choice of the empirical constants A, B and C in (27). Some authors proposed values in the range 150 – 200 for A and 1.7 – 4.0 for B as well as functional forms for these coefficients that depend on both porosity and Reynolds number, see overview in Plessis (2001). Vortwek and Brunn (1994) proposed to use values $A = 181$, $B = 2.01$ and $C = 0.96$ to estimate the pressure drop in randomly arranged packed beds. Franzen (1979) found that for $A = 164.97$, $B = 1.976$ and $C = 0.9$, equation (28) gives good estimate of the pressure drop for regularly arranged spheres in the cubic packing. The reason for variation in the constants was determined as the variations in particle geometry and orientation, as well as macroscopic properties of the packing. It should be noted that the Ergun equation was derived for densely packed beds, and is not expected to be valid for high void ratios. For that range (when porosity is smaller than 40 %), normally the Wen and Yu equation (Wen and Yu 1966) is used. However this equation significantly underpredicts the drag force at higher Reynolds numbers (Beetstra et al. 2007). Also the transition from the Ergun to the Wen and Yu equation is not a smooth function. For moving particles the situation becomes even more complicated since particles begin to interact with each other and may dissipate additional energy that affects the pressure drop. Therefore detailed mesoscale simulations that resolve the flow around each particle are needed to predict flow characteristics such as the pressure drop and volumetric flow rates in these systems.

As a first step, we validate our code against experimental and empirical data for both randomly and regularly arranged packings in different flow regimes and Reynolds numbers. First we consider a configuration that consists of monodisperse spheres of radius $R = 0.002m$ arranged in a cubic packing. We consider two flow regimes with Reynolds numbers $Re = 18$ and $Re = 900$. A constant inflow velocity field of $u = 0.5 \text{ m/s}$ was prescribed at the left boundary and open boundary conditions are set up at the right boundary. Other boundaries are chosen to be periodic. The porosity is $\epsilon = 47.64\%$. Figure 2 shows measured values of the friction coefficient Λ in different data sets that include

Ergun (1952); Franzen (1979); Hovekamp (2002); Martin et al. (1951); Vortwek and Brunn (1994). For small Re numbers, difference in the data sets for both random and cubic packings is small (Figure 2a). For high Re numbers, the friction coefficient is found to deviate by more than half an order magnitude in the existing empirical and experimental data (Figure 2b). Figure 3 shows flow patterns at different resolutions for $Re = 18$. We use Richardson extrapolation $f(h) = f_{exact} + Ch^p$ to estimate the convergence order using computed values of $f(h)$. Here f is a calculated parameter (pressure drop in our case), h is some measure of grid spacing, C is a constant, and p is the order of convergence. Based on the results of simulations we found that the convergence rate is close to a second order: 1.86 for $Re = 18$ and 1.8 for $Re = 900$ (Figure 4 and Figure 6, respectively). However the low Reynolds number flow requires less resolution to achieve a converged solution than the high Reynolds number flow. About 48 cells per particle are needed to describe flow behavior for $Re = 18$. In the case of $Re = 900$, the resolution needs to be twice higher to get the same order of error. This is mainly because of the flow separation and turbulent boundary effects that require finer grid resolutions to describe them adequately (Figure 5). In these simulations we do not use any turbulent model. However in future studies we may need to incorporate turbulent effects and additional drag terms to be able to simulate high Re number flow regimes with relatively modest resolution. Another option to improve the convergence rate is to revise the interpolation scheme for the velocity field in the mixed cells in (15). The grid resolution needed for a converged solution depends on the particular configuration and porosity. For densely packed beds, the flow characteristics are constrained by the maximum resolution available in the void space between particles. Therefore fine-resolution simulations are required to describe flow effects through these small void spaces for high Re numbers. The overall agreement with available data is very good (Figure 2).

The next example is flow in a periodic domain of randomly packed spheres of radius $R = 0.11283m$. Re number in this configuration is about 900 and the solid packing ratio is 60%. A superficial velocity is specified as $u = 0.008 \text{ m/s}$. The periodic random distribution of spheres is generated using Donev et al. (2005) algorithm. The pressure drop through the packed bed is illustrated in Figure 7. We perform a numerical analysis to estimate the convergence rate of the numerical solution. The error in the pressure drop decreases with approximately second order when refining the mesh (Figure 8). The difference between our converged solution and the Ergun equation (27) for coefficients $A = 150$ and $B = 1.75$ is found to be 7%. This is quite reasonable agreement

since the considered configuration with given porosity of 40% is a good representation of the cases investigated in the experiments by Ergun (1952). It should be noted that for loosely packed beds with porosity larger than 40% different random configurations may produce different results and deviations from Ergun values, this is discussed in Freund et al. (2003). In the next section we show how local inhomogeneities in the porosity distribution of randomly packed beds with the same global parameters (porosity, particle size) may influence the flow and transport properties.

Falling sphere

We further validate our code by investigating the sedimentation of a cylindrical particle in a Newtonian fluid. We consider sedimentation of a single two-dimensional disk with radius 1cm in a channel with dimension of $0.2\text{m} \times 0.4\text{m}$. The fluid density is $2000\text{kg}/\text{m}^3$ and the particle density is $2500\text{kg}/\text{m}^3$. The various Reynolds number dependent flow regimes are obtained by varying the fluid dynamic viscosity ν . We consider three cases here with $\nu = 3, 0.1$ and 0.002Pas . Gravity acceleration $9.8\text{m}/\text{s}^2$ acts in a negative y direction. The simulation is started at $t = 0\text{s}$ by dropping a particle at the center of the channel at 0.33cm depth. Different grid resolutions are considered ranging from 4 up to 256 grid cells per particle diameter.

Figure 9 shows vertical velocity snapshots and vorticity contours for $Re = 0.2$, $Re = 100$ and $Re = 2000$ correspondingly. Vorticity contours are plotted as an illustration of the different flow regimes which depend on the Re number. For the small Re number the flow is laminar and the disk reaches its terminal velocity relatively quickly (Figure 10a). With increasing the Re number a vortex forms behind the particle. For high Re numbers vortex shedding occurs that lead to horizontal deviation of the trajectory of the particle (Figure 9c). We found that the vortex shedding begins at $Re = 400$, which agrees well with experimental and theoretical findings (Achenbach 1974).

We also investigated the convergence rate for different Re numbers. The results agree well with the previous finding for the fixed array of spheres considered in the previous section. Whereas for small Re numbers, only few cells per particle diameter (around 6) are required to get a converged solution, for high Re numbers, about 32 grid cells are needed to represent the flow around the particle and the drag effects correctly (Figure 10, the right panel).

Particle flow through opening

The examples considered in the previous sections focused mainly on the flow through stationary packed beds. When the particles are not so constrained, the motion of the fluid leads to particle motion, and the moving particles interact with each other as well as with the fluid leading to complex flow patterns that depend significantly on the packing density and particle contact physics. For highly viscous fluids, viscous forces could prevent contact between particles, but for most applications, these forces are insufficient to prevent particle contacts. As it was mentioned in () to account for particle-particle interactions, separately applied collision forces similar to those used in the distinct element method (DEM) (Cundall et al. 1979) are implemented in the code. To validate the code performance and the particle contact physics algorithm, we performed simulations of a representative experiment conducted at UC Berkeley. The experimental configuration is shown in Figure 15. As shown, 2500 polypropylene pebbles, initially suspended in a vertical water column, were discharged through a narrow opening. The pebble reservoir was 40.6cm in diameter, while the diameter of the narrow opening has a diameter of 10.2cm . A conical region, with a 45 degree cone angle, connects the pebble reservoir to the narrow opening. The surface of the cone has been perforated to allow back flow of water into the reservoir as the pebbles were evacuated. The pebbles were all the same size, with a diameter of 2.5cm and a density of $0.843\text{g}/\text{cm}^3$. The bottom boundary of the container is closed, and a physical barrier is placed in the narrow opening to inhibit pebble motion such that, at the beginning of the experiment, both the pebbles and water are at rest.

The experiment is initiated by removing the physical barrier, thus allowing the pebbles to move upward through the water column. As the pebbles flow out of the lower reservoir, the evacuated volume is filled with water flowing into the reservoir through the porous conical section. In the simulations, the holes in the conical section were not modeled explicitly. Instead, the effect of the holes was taken into account by allowing the conical surface to physically constrain pebble motion while at the same time be transparent to fluid flow.

The comparisons between simulation results and experimental data are presented in Figures 11 and 12. The overall agreement is very good, though there are some discrepancies. Initially, simulation show that the pebbles move faster in the central region, and slower near the wall with some of the pebbles near the wall moving in a downward direction (see Figure 11a). This can be explained by the fact that the counter fluid flow is not distributed uniformly through the cone section. There could

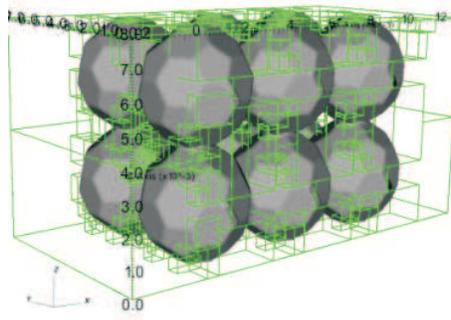


Figure 1: Geometry of AMR patches for simulation of fluid flow through the cubic array of spheres

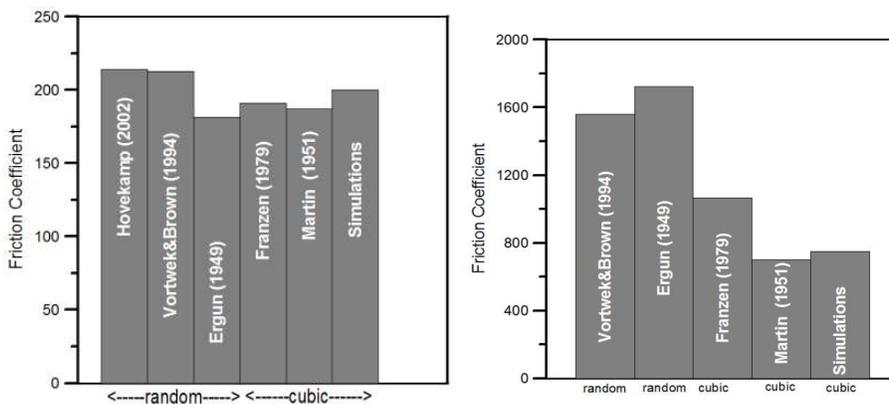


Figure 2: Comparison of the computed friction coefficient with available data for random and cubic packings at $Re = 18$ (left panel) and $Re = 900$ (right panel). Data from Hovekamp (2002), Vortwek and Brunn (1994), Ergun (1952), Franzen (1979) and Martin et al. (1951) are shown.

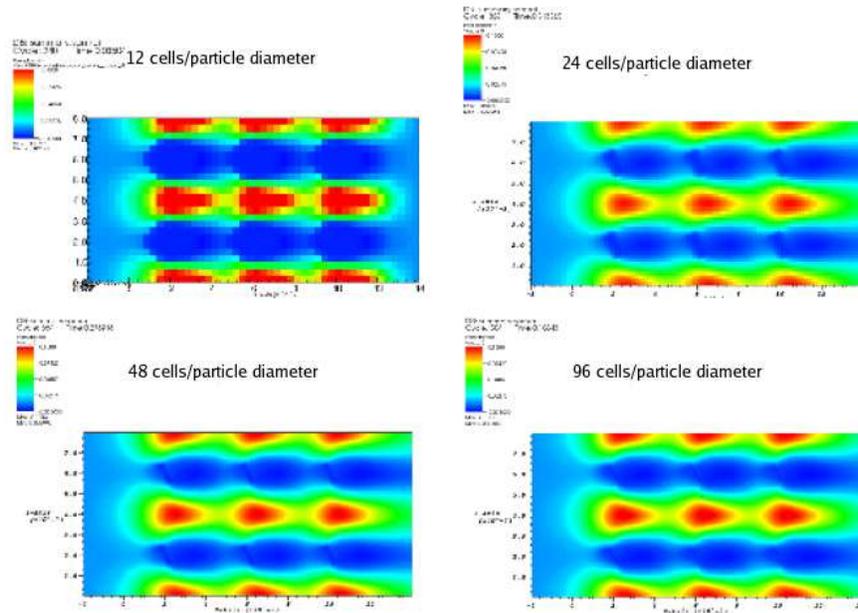


Figure 3: Horizontal velocities at the central slice for the cubic packing of spheres at $Re = 18$.

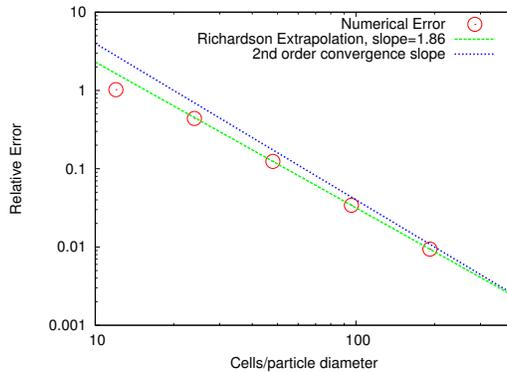


Figure 4: Convergence rate at low Reynolds numbers, $Re = 18$. Numerical error versus resolution is shown.

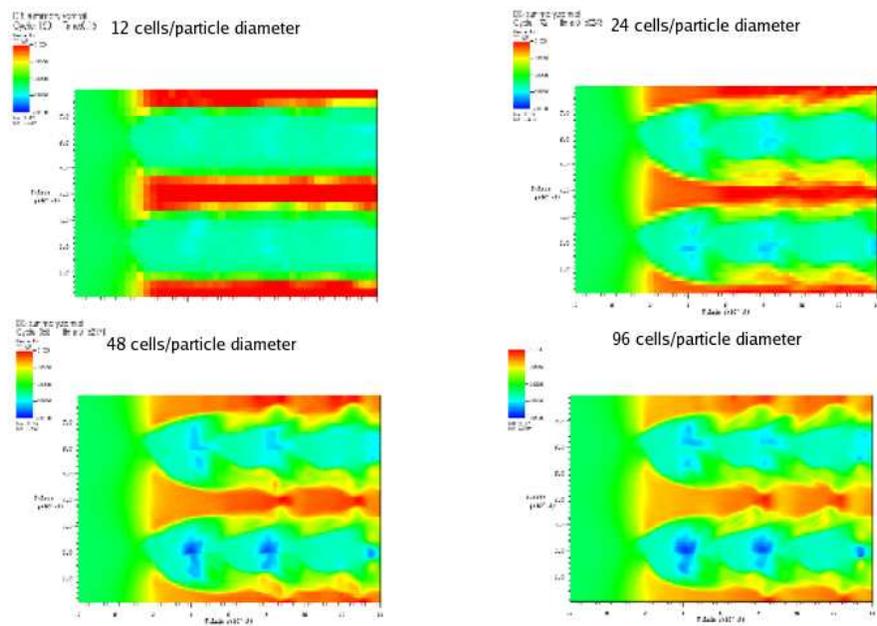


Figure 5: Horizontal velocities at the central slice for the cubic package of spheres at $Re = 900$

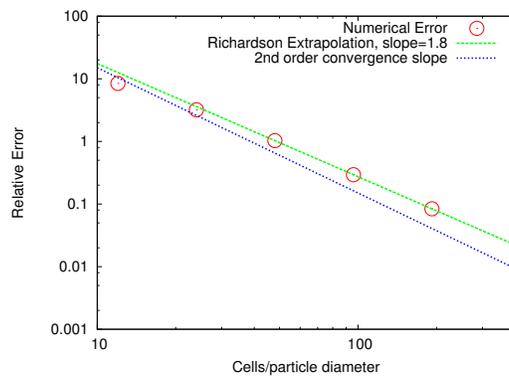


Figure 6: Convergence rate at low Reynolds numbers, $Re = 900$. Numerical error versus resolution is shown.

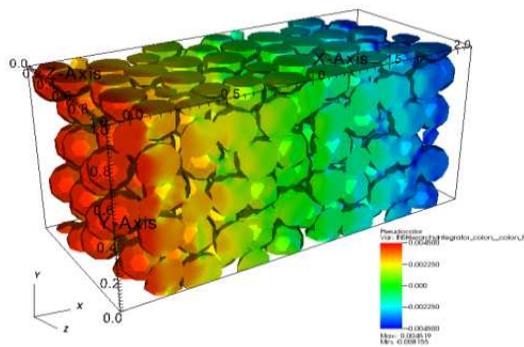


Figure 7: Pressure drop through the randomly packed bed.

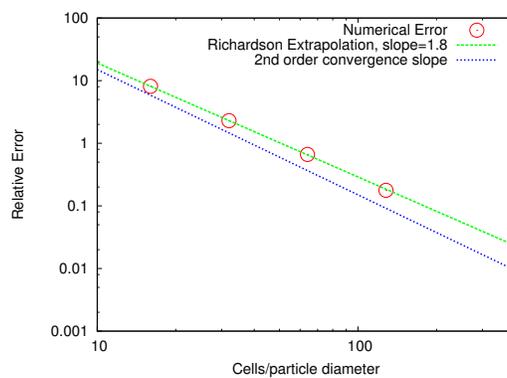


Figure 8: Convergence rate. Numerical error versus resolution is shown.

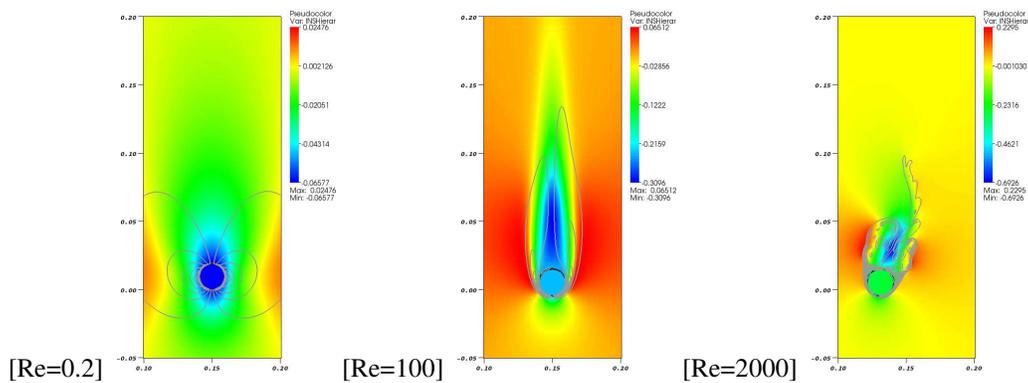
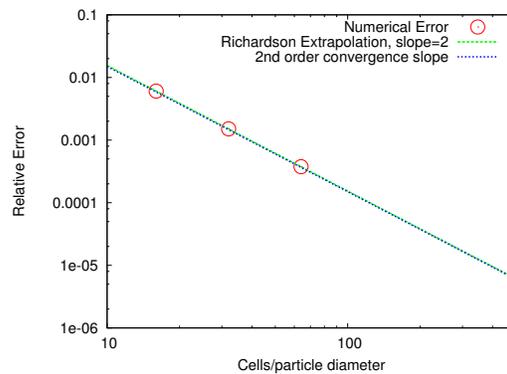
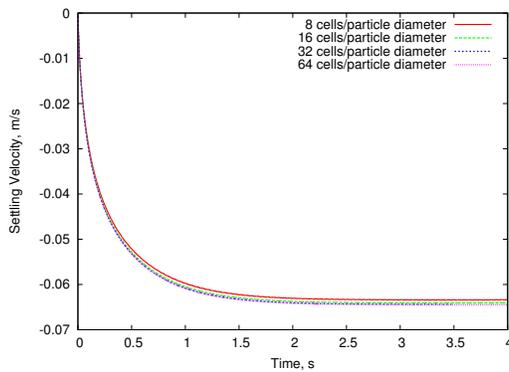
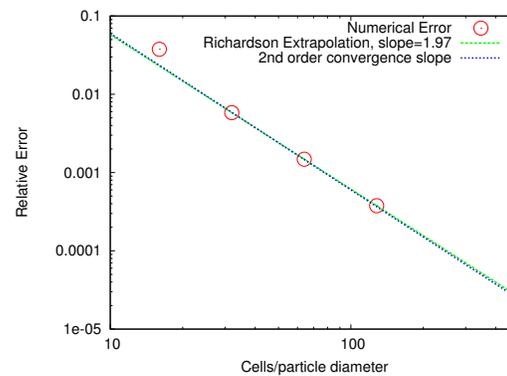
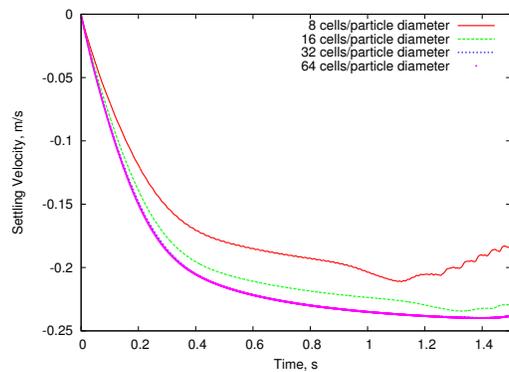


Figure 9: Numerical simulation of falling particle in a Newtonian fluid. A vertical velocity in the fluid is shown for different Reynolds numbers. Vorticity contours are plotted as an illustration of the different flow regimes

[Re=0.2]



[Re=100]



[Re=2000]

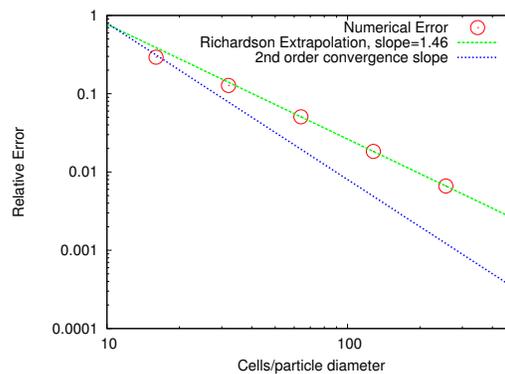
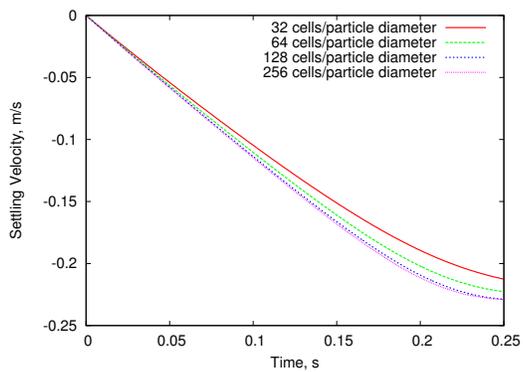


Figure 10: Settling velocity and convergence rate for $Re = 0.2$ (a), $Re = 100$ (b) and $Re = 2000$ (c)

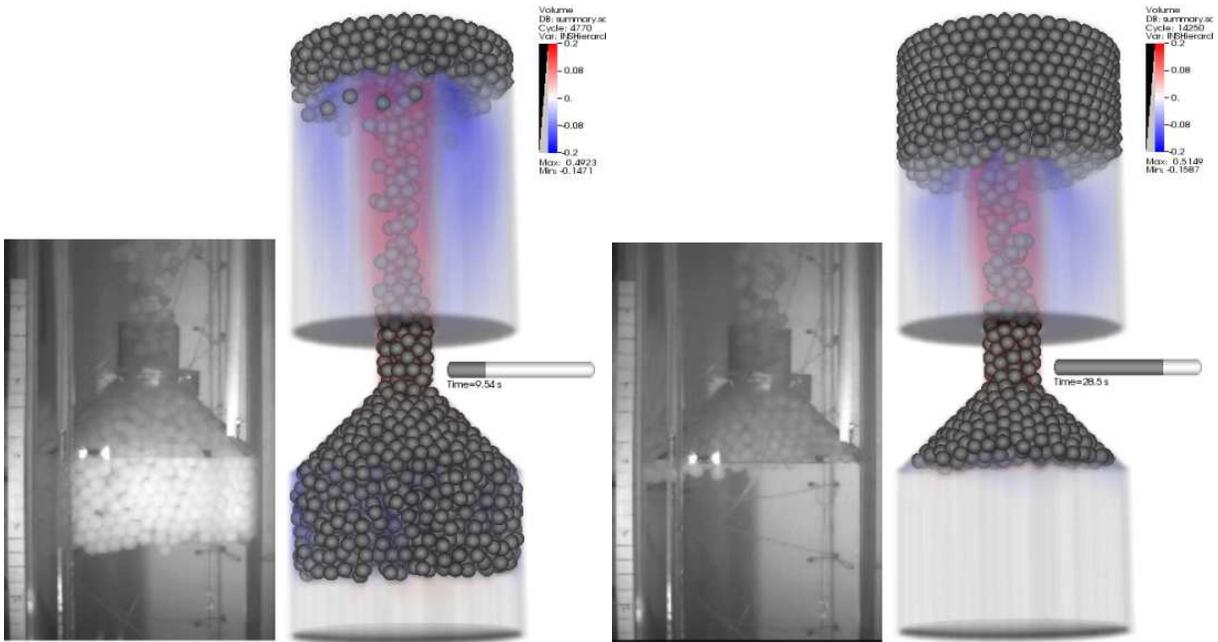


Figure 11: Comparison of the pebble flow in experiment and simulations at time 9.5 s (left panel) and 28.5 s (right panel)

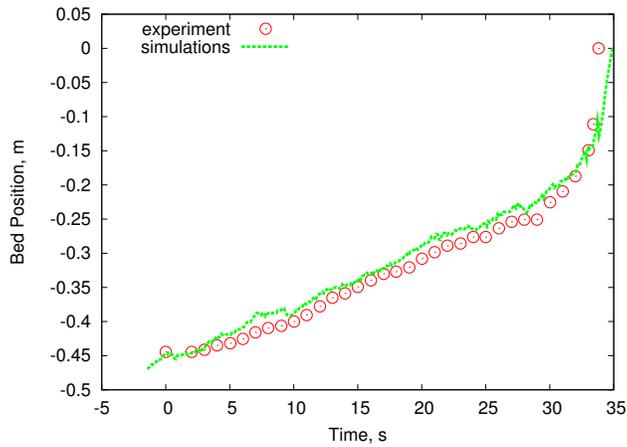


Figure 12: Comparison between experiment and simulations for the bed position below the chute outlet

be several reasons for this behavior. One reason may be that the holes in the conical section were not modeled explicitly. Another, more likely reason, is that our initial packing was not ideally conformed to the boundary of the container, thus leading to higher porosity near the skin of the container than in the interior of the pebble bed, thus leading to higher fluid flow rates near the outer boundary of the cylinder. However, it appears that those local fluctuations do not affect the integral characteristics of the flow field. A comparison of the simulated evolution of the bed bottom position with that observed during the experiment is shown in Figure 12. The bed bottom position was taken as the lowest pebble position in the central cylindrical slice of radius 5.1 cm .

Conclusions

We have presented a distributed Lagrange multiplier algorithm for particulate flows. The method is following ideas of Patankar et al. (2001); Sharma and Patankar (2005); Glowinski et al. (1999). The idea of the method is to use operator-splitting technique to solve the fluid equations in the entire domain first and then correct the flow inside the rigid domain using Lagrange multipliers. The parallel implementation of the algorithm is done using SAMRAI library. Previous work did not include collisions or base them on elastic forces that prevent particles from overlap. We extend this approach by incorporating DEM methods for inelastic, frictional contact forces. Following Patankar et al. (2001), the proposed numerical method does not include additional equations of motion for the particle translational and angular velocities. Code validation was done by comparing numerical results with known experimental and empirical data for a falling sphere in a Newtonian fluid, flow through the stationary packed beds, and pebble release through a narrow opening. We performed numerous convergence tests for different applications and found that for the low and moderate Re numbers the convergence of the method is close to the second-order. For the high Re numbers the convergence becomes slightly slower and could be explained by deficiencies in the flow representation near the rigid walls, in particular, the velocity interpolation in the mixed cells where both fluid and solid coexist and necessity to resolve turbulent structures in the fluid domain. The overall performance and accuracy of the code is very good and promises to be a valuable tool both for simulations of flow involving up to a few hundred thousand particles as well as for calibrating the phase-coupling relationships in two-fluid continuum simulation models.

Acknowledgements

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

- Achenbach E., 1974: Vortex shedding from spheres. *J. Fluid Mech.*, **62(2)**, 209-221.
- Almgren A.S., J.B. Bell, W. G. Szymczak, 1996: A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. *SIAM J. Sci. Comput.*, **17(2)**, 358-369.
- Balay S., K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang, 2004: PETSc Users Manual. Technical Report ABNL-95/11-Revision 2.1.5, Argonne National Laboratory, 2004.
- Beetstra R., M.A. van der Hoef, J.A.M. Kuipers, 2007: Numerical study of segregation using a new drag force correlation for polydisperse systems derived from lattice-Boltzman simulations. *Chemical Eng. Science*, **62**, 246-255.
- Bell J.B., P. Colella, L.H. Howell, 1991: An efficient second-order projection method for viscous incompressible flow. In 10th AIAA Computational Fluid Dynamics Conference, Honolulu, June 24-27, 1991.
- Berger J., J. Olinger, 1984: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.*, **53**, 484-494
- Colella P., 1985: A direct Eulerian MUSCL scheme for gas dynamics, *SIAM J. Comput.*, **6**, 104-117.
- Colella P., 1990: Multidimensional upwind method for hyperbolic conservation laws. *J. Comp. Phys.*, **87**, 171-200.
- Cundall P.A. and O.D.L. Strack, 1979: A discrete numerical model for granular assemblies. *Geotechnique*, **29**, 47-65.
- Donev A., F. H. Stillinger, and S. Torquato, 2005: Neighbor List Collision-Driven Molecular Dynamics Simulation for Nonspherical Particles. I. Algorithmic Details. *J. Comp. Phys.*, **202(2)**, 737-764.
- Ergun, S., 1952: Fluid flow through packed columns. *Chemical Engineering Progress* **48**, 89-94.

- Franzen P., 1979: Zum Einfluß der Porengeometrie auf den Druckverlust bei der Durchströmung von Porensystemen, I. Versuche an Modellkanälen mit variablem Querschnitt. *Rheol. Acta*, 18:392-423.
- Freund, H., T. Zeiser, F. Huber, E. Klemm, G. Brenner, F. Durst, G. Emig, 2003: Numerical simulations of single phase reacting flows in randomly packed fixed-bed reactors and experimental validation. *Chemical Engineering Science*, 58, 903-910.
- FLUENT 6.0 User's Guide Book, 2001.
- Gidaspow, D., 1994: *Multiphase flow and fluidization. Continuum and Kinetic theory descriptions*. Academic Press, NY.
- Glowinski, R., T.W. Pan, T.I. Hesla, D.D. Joseph, 1999: A distributed Lagrange multiplier fictitious domain method for particulate flow. *Int. J. Multiphase Flow*, 25, 755-794.
- Griffith B.E., R.D. Hornung, D.M. McQueen, C.S. Peskin, 2001: Parallel and adaptive simulation of cardiac fluid dynamics.
- Freund H., T. Zeiser, F. Huber, E. Klemm, G. Brenner, F. Durst, G. Emig, 2003: Numerical simulations of single phase reacting flows in randomly packed fixed-bed reactors and experimental validation. *Chemical Engineering Science*, 58, 903-910.
- Hertz H., 1882: Über die Berührung fester elastische Körper, *Journal of Reine und Angewandte Mathematik* 92, 156-171.
- Hornung, R.D., A.M. Wissink, S.R. Kohn, 2002: Managing complex data and geometry in parallel structured AMR applications. *Engineering with Computers*, 22(3-4), 181-195.
- Hovekamp, T.B., 2002: Experimental and numerical investigation of porous media flow with regard to the emulsion process. PhD thesis, 1-101.
- Marshall J.S., 2009: Discrete-element modeling of particulate aerosol flows. *J. Comp. Phys.*, **228**, 1541-1561.
- Martin, J.J., W.L. McCabe, C.C. Monrad, 1951: Pressure drop through stacked spheres: Effect of orientation. *Chemical Engineering Progress*, 47(2): 91-94. *Engineering with Computers*, 22(3-4), 181-195.
- Patankar N.A., 2001: A formulation for fast computations of rigid particulate flows. *Center Turbul. Res., Ann. Res. Briefs*, 185-196.
- Plessis J., 2001: Analytical Quantification of Coefficients in the Ergun Equation for Fluid Friction in a Packed Bed.!!YEAR AND JOURNAL!!!
- Sharma N. and N.A.Patankar, 2005: A fast computation technique for the direct numerical simulation of rigid particle flows. *J. Comp. Phys.*, **205**, 439-457.
- Syamlal M., W. Rogers and T.J.O. Brien, MFIx Documentation: Theory Guide U.S. Department of Energy (DOE), Morgantown Energy Technology Center, Morgantown, West Virginia (1993).
- Tsuji Y., T. Tanaka and T. Ishida, 1992: Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe. *Powder Technology*, 71, 239-250.
- Tsuji Y., T. Kawaguchi and T. Tanaka, 1993: Discrete particle simulation of two-dimensional fluidized bed. *Powder Technology*, 77, 79-87.
- Xu B.H., A.B. Yu, 1997: Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics. *Chemical Eng. Science*, 52, 2785-2809.
- Veeramani C., P.D. Mineev, K. Nandakumar, 2006: A fictitious domain formulation for flows with rigid particles: A non-Lagrange multiplier version. *J. Comp. Phys.*, **224**, 867-879.
- Vorwerk J., P.O. Brunn, 1994: Shearing effects for the flow of surfactant and polymer solutions through a packed-bed of spheres. *Journal of Non-Newtonian Fluid Mechanics*, 51(1), 79-95.
- Wen, C.Y. and Y.H. Yu, 1966: Mechanics of fluidization, *Chem. Eng. Prog. Symp. Ser* 62, 100-111.
- Zhu H.P., Z.Y. Zhou, R.Y. Yang, A.B. Yu, 2007: Discrete particle simulation of particulate systems: Theoretical developments. *Chemical Eng. Science*, 62, 3378-3396.