Computers & Fluids 48 (2011) 16–29

Contents lists available at ScienceDirect

Computers & Fluids

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

Mesoscale simulations of particulate flows with parallel distributed Lagrange multiplier technique

Y. Kanarska *, I. Lomov, T. Antoun

Lawrence Livermore National Laboratory, 7000 East Ave, L-286, Livermore, CA 94550, United States

ARTICLE INFO

Article history:
Received 12 June 2010
Received in revised form 21 December 2010
Accepted 11 March 2011
Available online 22 March 2011

Keywords:
Lagrange multiplier
Particulate flows

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. [16] and Patankar [30]. 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. [10] 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 particle 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 U.C. Berkeley Thermal Hydraulic Lab for pebble flow through a narrow opening.

1. Introduction

Particulate flows occur in a wide range of industrial applications and in nature. Clearly, it is 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 [42].

When the computational grid size is much larger then 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 [15] and various implementations are available, such as the commercial code FLUENT [14] and the DOE/NFTL code MFIX [35]. 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 multiphase 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. An overview of these

* Corresponding author.
E-mail address: kanarska1@llnl.gov (Y. Kanarska).

0045-7930/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2011.03.010

methods can be found in Zhu et al. [42]. The algorithm relies on the parameterizations of drag terms similar to the Ergun equation [12] for static packed beds, or the Wen–Yu equation for moving beds [41]. 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 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, 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.

We have selected the Lagrange multiplier technique following the ideas of Glowinski et al. [16] and Patankar [30] to model fluid-particle systems. As in most other numerical models for particulate flows, a fractional-step method is employed for the time integration procedure, where the fluid motion is calculated at the first time step and the rigid body projection of the intermediate velocity is calculated at the second time step [30,33,8,40]. The advantage of the method is that its finite element formulation permits the use of a fixed structured grid. This eliminates the need for quite expensive remeshing the domain, a necessity in the unstructured grid-based methods [24]. Another advantage of using fixed structured grid is in simultaneous efficient description of particle domain and collisions between particles in contrast to body fitted methods that become prohibitively expensive when particle collisions are considered [24]. We extend Distributed Lagrange Multiplier technique of Glowinski et al. [16] and Patankar [30] by implementing explicit force-displacement interactions for frictional inelastic particles similar to the DEM method [10] with some modifications using a volume of an overlapping region as an input to the contact forces. In the present work the contact is detected in the region where two particles overlap based on the volume of fraction function and does not require expensive contact searching algorithms. While this approach would not be applicable for "classical" DEM algorithms since the particle domain is not resolved in DEM, it is a natural fit for fluid-particle problems. Moreover, it is easily extendable to non-spherical objects which are hard to accommodate in a typical DEM approach. In addition, if the application domain has complex geometry (e.g. in pneumatic conveying systems) it needs to be represented in the numerical code. This is done using the volume of fraction technique that allows to treat different geometric objects of different shapes and sizes.

Parallel capabilities are important for improving efficiency of the numerical method. Despite on the large number publications devoted to the fictitious domain method its parallel implementation is rarely discussed. 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) technique and efficient representation of large amounts of solid objects. Rapid granular flows demonstrate lack of scale separations in many applications and will require multiscale approach that can be achieved using AMR. A parallel implementation of the method is based on the Lawrence Livermore Structured Adaptive Mesh Refinement Application Infrastructure (SAMRAI) library and allows local grid refinement at the solid interfaces. Eulerian methods with AMR and advanced interface tracking capabilities are well suited for problems where accurate tracking of solid-fluid and fluid-fluid interfaces are required. The purpose of AMR is to concentrate the computational work near the regions of interest (e.g. solid boundary or maximum vorticity region). When properly designed, AMR can significantly reduce the computational effort required to obtain a desired level of accuracy in the simulation. Griffith et al. [18,19] extended SAMRAI infrastructure for fluid-structure interaction problems using immersed boundary method [31]. This paper extends this framework using Distributed Lagrange Multiplier technique to describe fluid-solid interactions for the case of a large number of rigid particles. The convergence properties and scalability of numerical algorithm are investigated for different benchmark examples and applications. The paper is organized as follows. Introduction is presented in the first section. Numerical method, governing equations and algorithm are described in Section 2. Section 3 describes code validation against experiments and empirical data (Sections 3.1, 3.3 and 3.4) as well as some guidance for the future code applications (Section 3.2). The final section presents conclusions.

2. Numerical model

The method we used is based on the Distributed Lagrange Multiplier (DLM) technique of Glowinski et al. [16] and Patankar [30] 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 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 SAMRAI framework [22] developed at LLNL. This framework allows to track individual particle position on multiple cpu as well as do refinement resolution on structured grid near the areas of interest (e.g. solid-fluid interfaces, maximum vorticity zones etc.).

2.1. Collisionless governing equations

The idea of the Lagrange multiplier algorithm is based on the formulation of Glowinski et al. [16], Patankar [30], Sharma and Patankar [33]. 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 particle. The entire computational domain that includes both the fluid and the particles is denoted by $F \bigcup P$. The governing equations in the fluid domain can be written as:

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

$$\nabla\cdot\mathbf{u}=0, \text{ in } F\bigcup P \tag{2}$$

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

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

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