Models, algorithms and validation for opensource DEM and CFD-DEM

Article *in* Progress in Computational Fluid Dynamics An International Journal · June 2012

DOI: 10.1504/PCFD.2012.047457

5 authors, including:

Christoph Goniva
DCS Computing GmbH
34 PUBLICATIONS **355** CITATIONS
SEE PROFILE

Alice Hager
DCS Computing GmbH
12 PUBLICATIONS **193** CITATIONS
SEE PROFILE

Stefan Amberger
Johannes Kepler University Linz
4 PUBLICATIONS **176** CITATIONS
SEE PROFILE

Stefan Pirker
Johannes Kepler University Linz
117 PUBLICATIONS **679** CITATIONS
SEE PROFILE

Models, algorithms and validation for opensource DEM and CFD-DEM

Christoph Kloss*, Christoph Goniva, Alice Hager, Stefan Amberger and Stefan Pirker

All CD-Laboratory on Particulate Flow Modelling,
JKU Linz, Altenbergerstr. 69
4040 Linz, Linz, Austria
E-mail: christoph.kloss@jku.at
E-mail: christoph.goniva@jku.at
E-mail: alice.hager@jku.at
E-mail: stefan.amberger@jku.at
E-mail: stefan.pirker@jku.at
*Corresponding author

Abstract: We present a multi-purpose CFD-DEM framework to simulate coupled fluid-granular systems. The motion of the particles is resolved by means of the Discrete Element Method (DEM), and the Computational Fluid Dynamics (CFD) method is used to calculate the interstitial fluid flow. We first give a short overview over the DEM and CFD-DEM codes and implementations, followed by elaborating on the numerical schemes and implementation of the CFD-DEM coupling approach, which comprises two fundamentally different approaches, the unresolved CFD-DEM and the resolved CFD-DEM using an Immersed Boundary (IB) method. Both the DEM and the CFD-DEM approachare successfully tested against analytics as well as experimental data.

Keywords: DEM discrete element method; CFD-DEM; granular flow; particulate flow.

Reference to this paper should be made as follows: Kloss, C., Goniva, C., Hager, A., Amberger, S. and Pirker, S. (2012) 'Models, algorithms and validation for opensource DEM and CFD-DEM', *Progress in Computational Fluid Dynamics*, Vol. 12, Nos. 2/3, pp.140–152.

Biographical notes: Christoph Kloss is currently a Senior Research Associate at the Christian Doppler Laboratory on Particulate Flow Modelling at the Johannes Kepler University (JKU) in Linz, Austria. He is co-founder and managing partner of DCS Computing GmbH, Linz as well as founder and core developer of the 'CFDEMproject', where he is heading the development of the open source DEM code LIGGGHTS. He received his Diploma in Mechatronics in 2007 and his PhD in Computational Fluid Dynamics in 2011, both at the Johannes Kepler University in Linz.

Christoph Goniva is currently a Senior Research Associate at the Christian Doppler Laboratory on Particulate Flow Modelling at the Johannes Kepler University (JKU) in Linz, Austria. He is co-founder and managing partner of DCS Computing GmbH, Linz as well as co-founder and core developer of the 'CFDEMproject', where he is heading the development of the open source CFD-DEM code 'CFDEMcoupling'. He received his Diploma in Mechatronics in 2007 and his PhD in Computational Fluid Dynamics in 2011, both at the Johannes Kepler University in Linz.

Alice Hager studied Industrial Mathematics at Johannes Kepler University (JKU) in Linz/Austria. After finishing her master-thesis on 'parameter choice criteria for the regularisation of inverse problems' in 2010 she became research assistant at the Christian Doppler Laboratory on Particulate Flow Modelling at JKU. Currently she works on a PhD-thesis in the field of Open Source CFD-DEM (discrete element method) with OpenFOAM® and LIGGGHTS. At the moment her main focus lies on resolved methods for representing immersed bodies within a fluid.

Stefan Amberger studied Product Management at FH-OOE for a year, then changed his major to Technical Mathematics at Johannes Kepler University in Linz. After volunteering in Southeast Asia, he became Research Assistant at the Christian Doppler Laboratory on Particular Flow Modelling, attended the CSC Summer School on High Performance Computing and worked at Sandia National Laboratories in the US.

Stefan Pirker has received his academic education in Mechanical Engineering. His subsequent research activities focused on applied multiphase flows. In 2009 he was assigned head of a Christian-Doppler Laboratory on Particulate Flow Modelling which since then has grown to 20 scientific employees. In this specific institution research is organised in three groups: development of (a) discrete and (b) continuum model approaches as well as (c) their experimental validation.

1 Introduction

Many flows in nature as well as industry are particulate flows. During the last decades several strategies to numerically describe granular flows have been developed. Hereby, two main approaches for modelling the particulate phase can be found: the continuum approach and the discrete approach.

In a continuum approach the multitude of particles is considered as an artificial continuum and is based on the solution of the underlying conservation equations using Computational Fluid Dynamic (CFD) techniques (Gidaspow et al., 1992). Naturally, this approach does not account for the local behaviour of the individual particles. In general, such continuum models are derived for a specific flow regime. For rapid-granular flow, meaning a regime of binary collisions, the so-called kinetic theory for granular flow has been introduced and successfully applied in many cases. This approach stems from the kinetic theory of dense gases by Chapman and Enskog (e.g., Chapman and Crowling, 1970), which was then applied to granular flow by Ogawa (1978), using the concept of a granular temperature. The other extreme is the regime of quasi-static or slow granular flow, which is characterised by long-lasting multi-particle contacts. In this regime, a particle is typically stabilised by its neighbouring particles so that, as opposed to rapid granular flow, the particles flow collectively in this regime. In this regime, models related to the plasticity theory have been developed (e.g., Rao and Nott, 2008).

The discrete approach does not rely upon continuum mechanics. It rather describes the motion of each particle individually, with a special treatment of eventual collisions. The most important discrete model is the Discrete Element Method (DEM).

In industrial application, single phase 'dry' granular flow rarely occurs. In the vast majority of natural or industrial processes concerning granular materials, a secondary-fluid phase, such as air, is present and its effects like fluidisation (aeration of particles by gas injection) play an important role. In some cases, such as pneumatic conveying, the fluid-phase controls particle movement, but particle–particle interactions may still be an important issue that cannot be neglected, whereas in other applications, such as hopper discharge, the particle flow induces fluid flow.

A promising approach to model such coupled granular-fluid systems is a coupled CFD-DEM approach. Reviews on DEM and the CFD-DEM technique have been published by Zhu et al. (2007, 2008). Recently Kloss et al. (2009) presented a CFD-DEM coupling approach along with experimental validation for two commercial software packages for DEM and CFD, respectively. Yet, owing to shortcomings of the commercial software packages this approach is limited to shared memory machines.

In this paper, we will describe a CFD-DEM approach based on the Open Source software packages OpenFOAM® (OpenCFD Ltd., 2009) and LIGGGHTS (LIGGGHTS, 2011). We show the versatility of this coupling routine and the applicability to a variety of particle laden flows. Therefore, several applications along with validation work are presented. A basic version of this coupling is provided and maintained by the authors via a dedicated web page (CFDEM, 2011).

2 Model description

2.1 Discrete Element Method (DEM) implementation

The DEM was introduced by Cundall and Strack (1979). A very brief description of the method will be provided in this section. Further details on the contact physics and implementation issues are available in the literature (di Renzo and di Maio, 2004; Pöschel and Schwager, 2005; Zhu et al., 2007).

The strength of the DEM lies in its ability to resolve the granular medium at the particle scale, thus allowing realistic contact force chains and giving rise to phenomena induced by particle geometry combined with relative particle motion, such as particle segregation by percolation. Thereby, it is able to capture many phenomena, describe dense and dilute particulate regimes, rapid flow as well as slow flow and equilibrium states or wave propagation within the granular material.

Thanks to advancing computational power, the DEM has become more and more accessible lately. On state-of-the-art desktop computers, simulations of up to a million particles can be performed. On very large clusters, the trajectories of hundreds of millions of particles can be computed (e.g., SANDIA, 2011).

2.1.1 Governing equations

The DEM is a Lagrangian method, meaning that all particles in the computational domain are tracked by explicitly solving their trajectories. The force balance for the particle with index i reads

$$m_i \ddot{\mathbf{x}}_i = \mathbf{F}_{i,n} + \mathbf{F}_{i,t} + \mathbf{F}_{i,f} + \mathbf{F}_{i,b}, \tag{1}$$

$$I_i \frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = \mathbf{r}_{i,c} \times \mathbf{F}_{i,t} + \mathbf{T}_{i,r}, \tag{2}$$

where $\mathbf{F}_{i,n}$ is the normal particle–particle contact force, $\mathbf{F}_{i,t}$ is the tangential particle–particle contact force. $\mathbf{F}_{i,f}$ is the

force that the surrounding fluid phase may exert on the particles, which will not be discussed here further. Other body forces like gravity, electrostatic or magnetic forces are summarised to $\mathbf{F}_{i,b}$. $\mathbf{T}_{i,r}$ is an additional torque on the particle that can be used to model non-sphericity by means of a 'rolling friction' model (e.g., LIGGGHTS, 2011). Each particle is mathematically represented by a sphere, another geometrically well-defined volume or a combination of them. The translational and angular accelerations of a sphere are based on the corresponding momentum balances. In the soft-particle DEM, the particles are allowed to overlap slightly. The normal force tending to repulse the particles can then be deduced from this spatial overlap $\boldsymbol{\delta}_p$ and the normal relative velocity at the contact point, $\Delta\mathbf{u}_n$.

2.1.2 Particle–particle contact model

The simplest example is a linear spring-dashpot model, shown in Figure 1. In the frame of this model, the normal force is given by

$$\mathbf{F}_n = -k_n\boldsymbol{\delta}_n + c_n\Delta\mathbf{u}_n. \tag{3}$$

The magnitude of the tangential contact force can be written as:

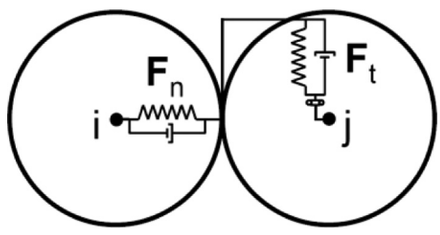
$$\mathbf{F}_t = \min\left\{\left|k_t\int_{t_{c,0}}^{t}\Delta\mathbf{u}_t\,\mathrm{d}t + c_t\Delta\mathbf{u}_t\right|,\ \mu\mathbf{F}_n\right\}, \tag{4}$$

where $\Delta\mathbf{u}_t$ is the relative tangential velocity of the particles in contact. k_n, k_t and c_n, c_t, are the normal and tangential spring and damping coefficients, respectively, and $\boldsymbol{\delta}_n$ is the normal overlap. This formula also holds in the case of a particle-wall contact. The tangential overlap is truncated to fulfil the Coulomb friction criterion with μ being the coefficient of friction.

The integral term represents an incremental spring that stores energy from the relative tangential motion, representing the elastic tangential deformation of the particle surfaces that happened since the time when particles touched at $t = t_{c,0}$. The second part, the dashpot, accounts for the energy dissipation of the tangential contact. The magnitude of the tangential force is limited by the Coulomb frictional limit, where the particles begin to slide over each other.

By varying k_n, k_t, c_n and c_t, as a function of the overlap, different kinds of contact models contact laws (linear and non-linear) can be implemented (see, e.g., di Renzo and di Maio, 2004; Pöschel and Schwager, 2005).

Figure 1 Simple spring-dashpot model



2.1.3 Particle–particle heat conduction model

A simple model for heat conduction similar to the model published by Chaudhuri et al. (2006) was implemented. The energy equation for the particle with index i reads

$$m_i c_{p,i}\frac{\mathrm{d}T_i}{\mathrm{d}t} = \sum_j q_{ij} + q_{\mathrm{src},i}, \tag{5}$$

where T_i and $c_{p,i}$ are the particle temperature and thermal capacity, and $q_{\mathrm{src},i}$ is the heat source (or sink) for the particle, e.g., stemming from chemical reactions. The heat flux q_{ij} stemming from conduction via inter-particle contacts is summed over all contacts (index j) of particle i. Under the assumption that the temperature is uniform within the particle, it can be written as

$$q_{ij} = h_{ij}(T_j - T_i), \tag{6}$$

where the heat transfer coefficient h_{ij} is calculated from the individual thermal conductivities k as follows:

$$h_{ij} = 4\frac{k_i k_j}{k_i + k_j}\sqrt{A_{c,ij}}. \tag{7}$$

The contact area between particles i and j is approximated as

$$A_{c,ij} = -\frac{\pi}{4}\frac{(c_{ij} - r_i - r_j)(c_{ij} + r_i - r_j)(c_{ij} - r_i + r_j)(c_{ij} + r_i + r_j)}{c_{ij}^{\,2}} \tag{8}$$

where

$$c_{ij} = \delta_{n,ij} - r_i - r_j. \tag{9}$$

This dependence on the normal overlap may lead to problems if the Young's modulus is chosen low in order to speed-up the DEM simulation. In this case, a correction to the heat transfer area has to be applied.

2.1.4 Contact detection algorithm

The concept of neighbour lists has been introduced by Verlet (1967). The idea is that a list of potential contacts is built periodically. Every time-step, this list is then checked and evaluations based on the actual contacts (e.g., computation of the particle–particle forces) are determined. This allows to a priori excluding pairs of particles that are too far away from each other to be in contact. A pair of (spherical) particles is included in the neighbour list if

$$\|\mathbf{x}_i - \mathbf{x}_j\| < r_i + r_j + s, \tag{10}$$

where s is the so-called Verlet or skin parameter that can be chosen freely within some bounds. It determines how long the list will stay valid. Assuming a constant time-step Δt and a maximum particle velocity with magnitude $v_{\max}$, the list is valid for

$$N_{\mathrm{verlet}} = \frac{s}{2v_{\max}\,\Delta t} \tag{11}$$

time-steps. Collision detection is based on this list for the next N_{verlet} time-steps before a new list is built. Usually, a binning approach such as the link-cell method via a grid decomposition is used to construct a neighbour list (e.g., Plimpton, 1995), i.e., to determine, which particle pairs are eligible for evaluation of Equation (4). The spacing of this binning and the Verlet parameter s are parameters to tune the run-time of the algorithm. It is obvious that for both the extreme limits for s, zero and infinity, the algorithm becomes useless because in the former case, no pair will be included in the list and in the latter case, all pairs in the system would be included, making the algorithm useless. No closed solution for this run-time optimisation is available since it not only depends on properties like particle size distribution, particle shape, the interaction force implementation, the flow regime, etc., but also on hardware specifications like CPU cache size.

2.2 Unresolved CFD-DEM approach

This approach is applicable to those cases where particle sizes are smaller than the computational grid, thus the particles are assumed to not completely fill a cell.

2.2.1 Governing equations

The motion of an incompressible-fluid phase in the presence of a secondary particulate phase is governed by the volume-averaged Navier–Stokes equations, which can be written as:

$$\frac{\partial \alpha_f}{\partial t} + \nabla \cdot (\alpha_f \mathbf{u}_f) = 0, \tag{12}$$

$$\frac{\partial (\alpha_f \mathbf{u}_f)}{\partial t} + \nabla \cdot (\alpha_f \mathbf{u}_f \mathbf{u}_f) = -\alpha_f \nabla \frac{p}{\rho_f} - \mathbf{R}_{pf} + \nabla \cdot \boldsymbol{\tau}. \tag{13}$$

Here α_f is the volume fraction occupied by the fluid, ρ_f is its density, $\mathbf{u}_f$ its velocity, and $\boldsymbol{\tau} = \nu_f \nabla \mathbf{u}_f$ is the stress tensor for the fluid phase. $\mathbf{R}_{pf}$ represents the momentum exchange with the particulate phase, which is calculated for each cell where it is assembled from the particle-based drag forces.

For solving above-mentioned equations, a pressure-based solver using PISO pressure velocity coupling (Issa, 1986) is used.

The coupling routine consists of several steps:

- the DEM solver calculates the particles positions and velocities
- the particles positions and velocities are passed to the CFD solver
- for each particle, the corresponding cell in the CFD mesh is determined
- for each cell, the particle volume fraction as well as a mean particle velocity is determined
- on the basis of the particle volume fraction, the fluid forces acting on each particle are calculated
- particle–fluid momentum exchange terms are assembled from particle-based forces by ensemble averaging over all particles in a CFD cell
- the fluid forces acting on each particle are calculated and sent to the DEM solver and used within the next time step
- the CFD solver calculates the fluid velocity taking into account local volume fraction and momentum exchange
- additional equations, such as species concentration can optionally be evaluated
- the routine is repeated from Equation (1).

Usually, DEM-time steps need to be set at least an order of magnitude smaller than CFD time steps owing to high-particle collision dynamics and requirements on maximum particle overlap for particle collision modelling, when using soft-sphere approach. To allow a flexible coupling scheme, the CFD and DEM time-steps can be set independently of each other. Additionally, a coupling interval can be chosen by the user to adapt to the amount of physical coupling between the phases.

For special cases (e.g., when calculating radiation or chemical reactions on the CFD side), it might be desirable to use CFD time steps smaller than the coupling time-step. In these cases it is possible (and advisable) to temporally under-relax the momentum exchange fields on the CFD side. This helps to reduce discontinuities in the exchange fields and thus improves convergence behaviour.

From the above-mentioned list of calculations necessary to realise the coupling it becomes obvious that for the calculation of the volume fraction (step 4) it is necessary to interpolate the particles' volumes, a Lagrangian property, from the DEM side to the volume fraction field, which is defined on the fixed Eulerian grid of the CFD simulation.

A very simplified approach for this transformation is to simply sum up the volume of those particles whose centres are located in a CFD cell. We will further refer to this approach as 'centreed' volume fraction calculation. This approach can lead to erroneous results owing to artificially inhomogeneous volume fraction field when the particle size approaches cell size.

Another approach, further referred to as 'divided' volume fraction calculation, is to resolve a particle by a series of distributed marker points and thus evenly apportion the particle's volume to all cells being (partly) covered. This on one hand helps to smoothen exchange fields without artificially enlarging spatial influence of a particle, but on the other hand its applicability is limited. When particle size approaches cell size, this method is no longer applicable.

One possible approach to overcome this problem was proposed by Link et al. (2005), where the spatial region of influence is artificially enlarged and thus exchange fields are smeared out. Being very efficient for structured grids this approach is hardly applicable on arbitrary unstructured grids used within this study.

A very similar approach was tested by the authors. It will further be referred as 'big particle' volume fraction calculation. For this approach the particles are treated as essentially bigger, but porous particles having the original volume. The region of influence of one particle (essentially the particle's contribution to the momentum exchange term,

Eq. 13) is thus extended to cells that are neighbours of the cell(s) where the particle is actually located, making the algorithm numerically more stable. The determination of those cells can be done recursively and in a way, which is suitable also for arbitrary-structured grids.

Once the particle volume fraction is calculated it is possible to evaluate each particle's contribution to particle–fluid momentum exchange, which is mostly established by means of a drag force depending on the granular volume fraction.

2.2.2 Momentum exchange model

For numerical reasons the momentum exchange term is split-up into an implicit and an explicit term using the cell-based ensemble averaged particle velocity $\langle \mathbf{u}_p \rangle$:

$$\mathbf{R}_{pf} = \mathbf{K}_{pf}(\mathbf{u}_f - \langle \mathbf{u}_p \rangle), \tag{14}$$

where

$$\mathbf{K}_{pf} = \frac{\sum_i \mathbf{F}_d}{V_{\text{cell}} \cdot \left| \mathbf{u}_f - \langle \mathbf{u}_p \rangle \right|}. \tag{15}$$

For the calculation of $\mathbf{K}_{pf}$ many different drag correlations have been proposed during the recent years (Zhu et al., 2007; Tsuji et al., 2008; Kafui et al., 2002). A widely used model was proposed by Gidaspow et al. (1992), which is a combination of the Wen and Yu (1966) model and the Ergun (1952) equation.

For $\alpha_f > 0.8$ momentum exchange is calculated as:

$$\mathbf{K}_{pf} = \frac{3}{4} C_d \frac{\alpha_f (1-\alpha_f) \left| \mathbf{u}_f - \mathbf{u}_p \right|}{d_p} \alpha_f^{-2.65}, \tag{16}$$

$$C_d = \frac{24}{\alpha_f \mathrm{Re}_p} \left[1 + 0.15 (\alpha_f \mathrm{Re}_p)^{0.687} \right], \tag{17}$$

$$\mathrm{Re}_p = \frac{\left| \mathbf{u}_f - \mathbf{u}_p \right|}{\nu_f} d_p, \tag{18}$$

whereas for $\alpha_f \leq 0.8$ the Ergun equation is used:

$$\mathbf{K}_{pf} = 150 \frac{(1-\alpha_f)^2 \nu_f}{\alpha_f d_p^2} + 1.75 \frac{(1-\alpha_f) \left| \mathbf{u}_f - \mathbf{u}_p \right|}{d_p}. \tag{19}$$

More recently a drag relation based on Lattice-Boltzmann simulations was proposed by Koch and Hill (2001).

Besides the drag force resulting from a relative velocity between the particle and the fluid, other forces, neglected in this work, may be relevant too. These may stem from the pressure gradient in the flow field (pressure force), from particle rotation (Magnus force), particle acceleration (virtual mass force) or a fluid velocity gradient leading to shear (Saffman force). The modular implementation of the CFD-DEM coupling allows implementing additional forces easily and superposing all forces acting on a particle.

2.2.3 Convective heat-transfer model

The CFD-DEM coupling allows accounting for convective heat transfer between the particulate phase and the fluid phase. An additional scalar transport equation for the temperature is solved:

$$\frac{\partial T_f}{\partial t} + \nabla \cdot (T_f \cdot \mathbf{u}_f) = \nabla \cdot (\kappa_{\text{eff}} \nabla T_f) + S_T. \tag{20}$$

Following Li and Mason (2000) the source term for the temperature equation can be calculated from the particle Nusselt number. For $\mathrm{Re}_p < 200$, it reads

$$\mathrm{Nu}_p = 2 + 0.6 r_p^n \mathrm{Re}_p^{1/2} \mathrm{Pr}^{1/3}, \tag{21}$$

for $200 < \mathrm{Re}_p < 1500$ we have

$$\mathrm{Nu}_p = 2 + 0.5 r_p^n \mathrm{Re}_p^{1/2} \mathrm{Pr}^{1/3} + 0.02 r_p^n \mathrm{Re}_p^{0.8} \mathrm{Pr}^{1/3}, \tag{22}$$

and for $1500 < \mathrm{Re}_p$

$$\mathrm{Nu}_p = 2 + 0.000045 r_p^n \mathrm{Re}_p^{1.8}. \tag{23}$$

Re_p denotes the particle Reynolds number and Pr the Prandtl number. For the exponent n the value $n = 3.5$ was found to give good results. With the heat transfer coefficient

$$h = \frac{\lambda \mathrm{Nu}_p}{d_p}, \tag{24}$$

where λ denotes the thermal conductivity the rate of heat transfer, we can calculate

$$q_p = h A_p (T_f - T_p). \tag{25}$$

For the source term S_T, we find

$$S_T = -\frac{q_p}{\rho_f C V_{\text{cell}}}. \tag{26}$$

2.3 Resolved CFD-DEM approach

This approach is applicable to those cases where particle sizes are bigger than the computational grid, thus the particles are assumed to cover multiple (e.g., at least ten) cells. In this approach the particulate phase is represented by a fictitious domain method (Patankar et al., 2000) where only one velocity and pressure field for both phases exists. Those regions covered by a particle have the same velocity as the particle itself.

2.3.1 Fluid flow

The governing equations of the fluid domain are the incompressible Navier–Stokes equations, which guarantee the conservation of momentum and mass, combined with boundary and initial conditions. These equations hold on

the whole domain $\Omega = \Omega_F + \Omega_P$ (see Fig. 2). The equations of motion for the particles and the surrounding fluid as well as the boundary and initial conditions can be summarised as follows (Shirgaonkar et al., 2008):

$$\rho_f \frac{\partial \mathbf{u}_f}{\partial t} + \rho_f (\mathbf{u}_f \cdot \nabla)\mathbf{u}_f = -\nabla p + \mu \nabla^2 \mathbf{u}_f \quad \text{in } \Omega_F, \tag{27}$$

$$\nabla \cdot \mathbf{u}_f = 0 \quad \text{in } \Omega_F, \tag{28}$$

$$\mathbf{u}_f = \mathbf{u}_\Gamma \quad \text{on } \Gamma, \tag{29}$$

$$\mathbf{u}_f = \mathbf{u}_p \quad \text{and} \quad \sigma \cdot \hat{\mathbf{n}} = t_{\Gamma_P} \quad \text{on } \Gamma_P, \tag{30}$$

$$\mathbf{u}_f(x,\, t=0) = \mathbf{u}_0(x) \quad \text{in } \Omega_F. \tag{31}$$

The time-discretised form of (Eq. 25) is given by

$$\rho_f \frac{\hat{\mathbf{u}}_f - \mathbf{u}_f^{(n-1)}}{\Delta t} + \rho_f \left(\mathbf{u}_f^{(n-1)} \cdot \nabla\right)\mathbf{u}_f^{(n-1)} = -\nabla \tilde{p} + \mu \nabla^2 \mathbf{u}_f^{(n-1)}. \tag{32}$$

Here, $\tilde{p}$ denotes an estimated value for the pressure, which will later on be corrected (PISO), $\hat{\mathbf{u}}_f$ is the interim solution, $\mathbf{u}_f^{(n-1)}$ is the solution at the previous time step. For correcting $\tilde{p}$ we use a Poisson-equation for the pressure.

The next step consists of correcting the velocity in the particle domain to the velocity taken from the DEM-data or an input-file, which yields the current velocity $\hat{\mathbf{u}}_f$. Technically speaking this is equivalent to adding a force term f to the Navier–Stokes equations, which fulfils

$$\rho_f \frac{(\tilde{\mathbf{u}}_f - \hat{\mathbf{u}}_f)}{\Delta t} = f. \tag{33}$$

The divergence-free condition for the velocity over the whole domain is violated now. To fix this, the following correction-operation is applied to the velocity:

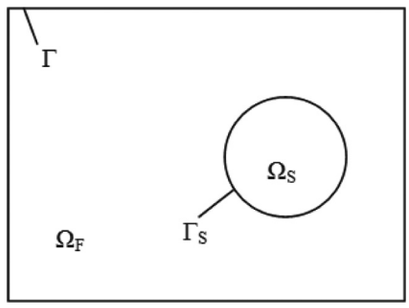
$$\mathbf{u}_f^{(n)} = \tilde{\mathbf{u}}_f - \nabla \phi. \tag{34}$$

Here, $\mathbf{u}_f^{(n)}$ is the final solution, ϕ a scalar field. As one wants to end up with $\nabla \cdot \mathbf{u}_f^{(n)} = 0$, we get the following Poisson-equation for ϕ:

$$\nabla^2 \phi = \nabla \cdot \tilde{\mathbf{u}}_f. \tag{35}$$

Correcting the velocity as described earlier is equivalent to adding an additional force-term to the Navier–Stokes equations.

Figure 2 Sketch of particle domain and fluid domain



2.3.2 CFD-DEM coupling

Generally, the force on a particle is given by

$$\int_{\Omega_p} \eta(x,\, y,\, z,\, t)\mathrm{d}V = \int_{\Omega} \eta(x,\, y,\, z,\, t)\xi(x,\, y,\, z,\, t)\,\mathrm{d}V, \tag{36}$$

where,

$$\xi = \begin{cases} 1 & \text{if } (x,y,z) \in \Omega_S \\ 0 & \text{else} \end{cases}. \tag{37}$$

η consists of the viscous component

$$\eta = \rho \nu_f \nabla^2 \mathbf{u}_f^{(n-1)}, \tag{38}$$

and the pressure-component

$$\eta = -\nabla p^{(n-1)}. \tag{39}$$

However, this method is computationally expensive as the the ratio of particle size to CFD cell size should be sufficiently small (e.g., 0.1). Thus, this method allow detailed resolution of the flow situation for small-scale fluid-particle systems.

The coupling routine consists of several steps:

- the DEM solver calculates the particles positions and velocities
- the particles positions and velocities are passed to the CFD solver
- a first fluid flow field is calculated
- for each particle, the corresponding cells in the CFD mesh are determined
- particle velocity is corrected in those cells covered by the particle
- the fluid forces acting on each particle are calculated and sent to the DEM solver and used within the next time step
- the flow field is corrected to be divergence free
- additional equations, such as species concentration, can optionally be evaluated
- the routine is repeated from Equation (1).

3 Solver infrastructure

3.1 DEM solver

LIGGGHTS is a parallel C++ DEM code based on the Molecular Dynamics (MD) code LAMMPS (Plimpton, 1994; lammps.sandia.gov) distributed by the authors via a dedicated web page (www.cfdem.com).

3.1.1 Parallelisation

This section outlines the MPI parallelism without dynamic-load balancing as implemented in LAMMPS (Plimpton, 1994) and LIGGGHTS. Basically, the domain is decomposed into several MPI processes using an orthogonal processor grid. Each of the sub-domains contains an owned region,

as well as a so called ghost region (also often referred to as halo or corona). The particles residing in the owned domain are called owned particles. Each process is responsible for integrating the trajectories of his owned particles. The ghost particles are the particles owned by neighbouring processes, which are more likely to interact with particles owned by neighbouring sub-domains. On the basis of the particle radii and the skin parameters, we can identify, which particles of each process have to be copied as ghost particles to the respective neighbouring processes. Subsequently, multiple instances of a particle can exist, yet it is clear at each time-step, which processor owns a particle, i.e., which processor is responsible for trajectory calculation. During the simulation, the positions, velocities and other properties of the owned particles are communicated to their ghost counterpart(s) each time-step before the actual force calculation is executed. The forces that a ghost particle experiences by owned particles is then communicated back, so that the processor that owns the particle can compute the correct trajectory, including all forces properly. Particles can migrate to other processors in the course of a simulation run so both the group of owned particles as well as the group of ghost particles for a process is not a static assignment, but will in general vary over time. Using this approach, excellent scalability has been shown for LAMMPS for the case of well-balanced domains, see Figure 3.

A second example shows the scalability of LAMMPS and LIGGGHTS for the simulation of a rotating drum using 100,000 particles. The drum is 1.09 m long and 0.73 m in diameter. The particles radius is 6.204 mm and the drum rotation speed is 30 RPM (results shown in Fig. 4) two different cases are shown. The drum is partly (~30%) filled with particles. In case 1, the code determines the domain decomposition by minimising the inter-processor communication area, irrespective of the spatial particle distribution. In case 2, the processors were manually assigned along the drum axis, resulting in better load-balance. It can be seen that case 2 yields a far better scalability, with a parallel efficiency of 85% for 32 processors, compared with a parallel efficiency of 32% for 32 processors in case 2.

Figure 3 Scaled-size LAMMPS scalability for a number of clusters at Sandia National Labs for a well-balanced chute flow simulation, from http://lammps.sandia.gov/bench.html#chute

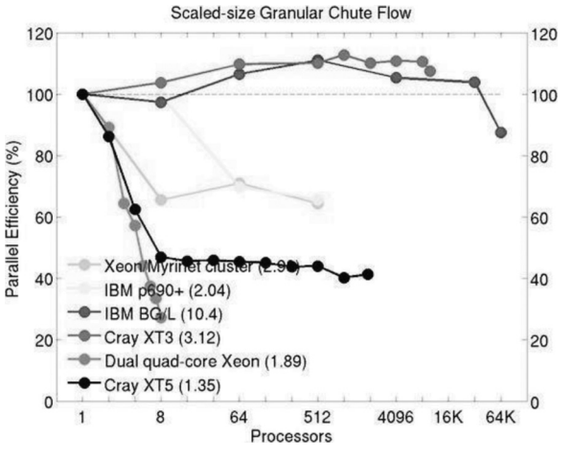
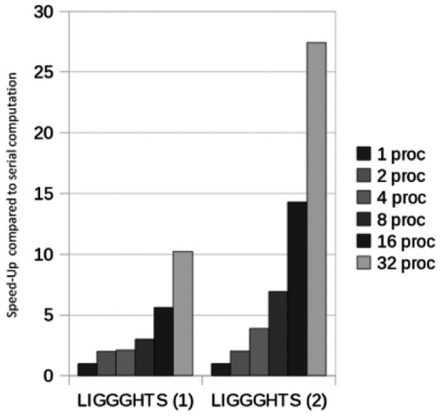


Figure 4 Strong (fixed-size) LAMMPS/LIGGGHTS scalability for simulation of a rotating drum with 100,000 particles with sub-optimal (case 1) and optimal domain decomposition (case 2)



We can thus conclude that a balanced particle distribution over the domains is very important to maintain code scalability. However, manual assignment of the domain decomposition (as performed in this example) may not always help to circumvent load imbalance. For this reason, we propose a dynamic load-balancing procedure.

3.1.2 Load-balancing

In general, load balancing is a technique to distribute workload evenly across two or more computers, network links, CPUs or other resources, to get optimal resource utilisation combined with maximised throughput, which is a topic widely interesting to computational mechanics, e.g., Hendrickson and Devine (2000).

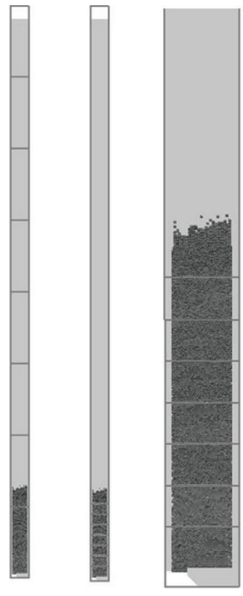
In the case of DEM simulations, the workload of each processor is obviously roughly proportional to the number of owned particles. The standard approach used by LAMMPS is to decompose the simulation space once at the start of the simulation into equally large parts based on a 3D grid of processors, which will in general lead to an unequal number of particles in each processor's sub-domain. Also, a static one-time domain decomposition that is initially well-balanced can lead to load imbalance for many granular flow situations during the simulation run, e.g., in the case of hopper discharge, where the spatial particle distribution will vary over time. Also, in the case of segregation simulations, where smaller particles have the tendency to accumulate in certain regions in the simulation domain, load imbalance is to be expected.

A simple approach that was implemented in LIGGGHTS to load balance DEM simulations is:

Perform the decomposition with orthogonal cuts in x-, y- and z-direction so that these cuts divide the particle entity equally.

A test case of a silo partly filled with mono-disperse granulate has been used to test the load balancing approach. Figure 7 shows snapshots of a largely empty silo. Using the standard approach with 8 processors, the simulation domain would be split up evenly such as shown on the lhs of Figure 5. The dynamic load balancing feature periodically calculates decomposition, such as shown on RHS of Figure 5, starting

Figure 5 Domain decomposition shown for a silo test case without dynamic load balancing (left) and with dynamic load balancing (middle and right) for 8 processors

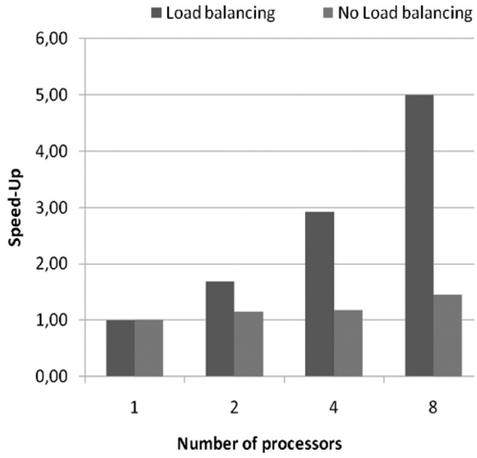


from the decomposition shown on the LHS of Figure 5, showing a close-up of the domain decomposition. Figure 6 shows the scalability results for 20,000 time-steps of simulation with the material being in rest within the silo. It can be seen that the scalability is improved significantly, with a parallel efficiency of 62% for a simulation with eight processors. The authors assume that the lower scalability compared with Figures 3 and 4 are owed to the slower inter-connect of the machine where this test case was run on (which was comparable with the 'Intel Xeon' desktop workstation shown in magenta in Fig. 3).

3.2 Coupled CFD-DEM solver

The method presented here treats the fluid and particle calculations in two strictly separated codes. This allows for taking advantage of independent code development on either side.

Figure 6 Speed-up for a silo test case with and without dynamic load balancing for simulations with 1–8 processors



The interaction is realised by exchange fields being evaluated in a predefined time interval, where the codes work in a sequential manner. Both the CFD and the DEM code do their calculations in parallel using Message Passing Interface (MPI) parallelisation. Also, data exchange between the codes is realised using MPI functionality.

The CFD-DEM approach described earlier was implemented within an open-source environment. The DEM simulations are conducted by the DEM code LIGGGHTS (LIGGGHTS, 2011) and the CFD simulations are conducted by a solver realised within the open-source framework of OpenFOAM® (OpenCFD Ltd., 2009). The coupling routines are collected in library providing a modular framework for CFD-DEM coupling with the C++ codes LIGGGHTS and OpenFOAM®. Both a selection of coupling routines as well as example solvers are provided at a dedicated web page maintained by the authors (www.cfdem.com).

4 Validation cases

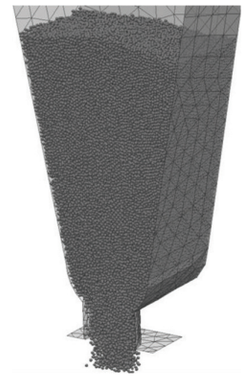
4.1 DEM: Hopper flow

Here, we present the validation results for a series of 28 experimental set-ups, each conducted with mono-disperse spherical glass beads with different particle diameters ranging from 2 mm to 4 mm and variable hopper-orifice diameters, ranging from 14 mm to 38 mm. The hopper used is equivalent to the one described by Kloss et al. (2010).

For the simulations, LIGGGHTS was run in multi-partition mode with a total of 36 processors, split up into 9 partitions with 4 processors each. The particle number was ranging from 50,000 to 400,000 for the different cases. 500,000 time-steps were simulated for each case to obtain the steady-state flow-rate. A simulation snapshot is shown in Figure 7. All the 28 simulations were controlled via one single input file via scripting, and the jobs were assigned to the partitions until all jobs were finished. With this technique, the results for all the 28 cases could be obtained within 24 h of calculation time on 36 processors on our cluster.

The results are shown in Figure 8 and it can be seen that for most of the cases, the simulated and measured mass-flow rates are in good agreement.

Figure 7 Snapshot for a hopper discharge simulation



4.2 DEM: heat conduction in packed bed

This example case calculates the conductive heat transfer in a fixed granular bed consisting of 11,600 particles with a radius of 5 mm. The bed is 1 m in height, width and depth are both 0.1 m. Thermal conductivity of the particles was set to 5 W/(K m), thermal capacity to 0.1 J/(kg K) and particle density to 2500 kg/m³. Initially, the bed temperature is 263 K throughout the bed. At $t = 0$ s, heating starts at the bottom of the bed ($z = 0$ m). Three seconds of real time were simulated, where one second of real time for the DEM simulation took 5 min on 4 cores of a desktop computer. The results were compared with a continuum model, corresponding to equation 18 without the convective transport term on the lhs. However, it must be noted that the continuum model does not reflect the dependency of particle–particle contact area on the Young's modulus. The results shown in Figure 9 correspond to a Young's Modulus of 2.2e6 Pa for the DEM calculation.

4.3 Unresolved CFD-DEM: heat conduction and convection

In this example case, the convective-heat transfer between the granular bed and a fluid is calculated in addition to conductive heat transfer in the bed. The dimensions of the bed are as described in Section 4.2, with the difference that the bed consists of 8650 particles with a diameter of 0.011 m. The fluid and granular properties are those of air and glass at 20°C. While the particle's temperature is kept constant at 600 K, the air enters from the bottom at 263 K and 1 m/s. The results are shown in Figure 10.

From the global fluid-particle heat flux, the Nusselt number is calculated with

$$\dot{q} = \dot{m} C_p (T_{\text{out}} - T_{\text{in}}),$$
$$h = \frac{\dot{q}}{n_p d_p^2 \pi \dfrac{T_{\text{out}} + T_{\text{in}}}{2}}, \tag{40}$$
$$Nu_p = \frac{h d_p}{\lambda},$$

Figure 8 Comparison of simulated (DEM) and measured hopper mass-flow rates for 28 experimental set-ups

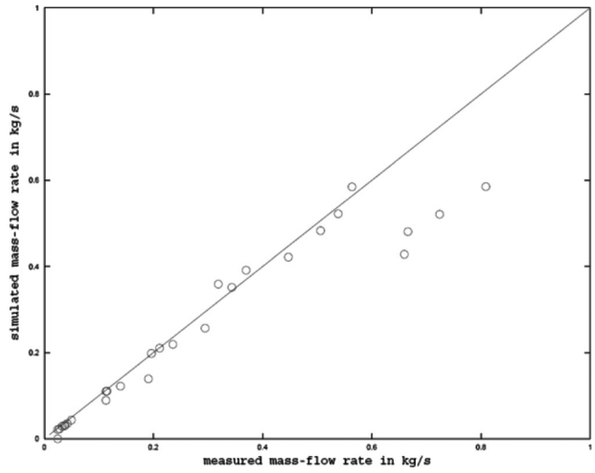


Figure 9 Comparison of DEM and finite volume simulation for heat transfer in a packed bed, curves representing temperature distributions at $t = 1$ s, $t = 2$ s and $t = 3$ s

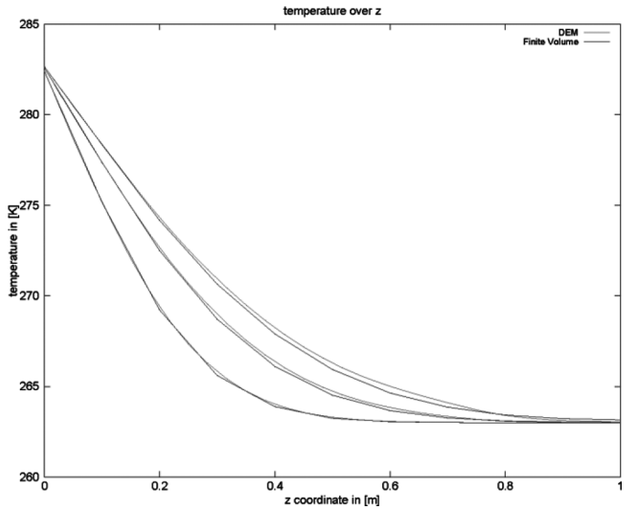
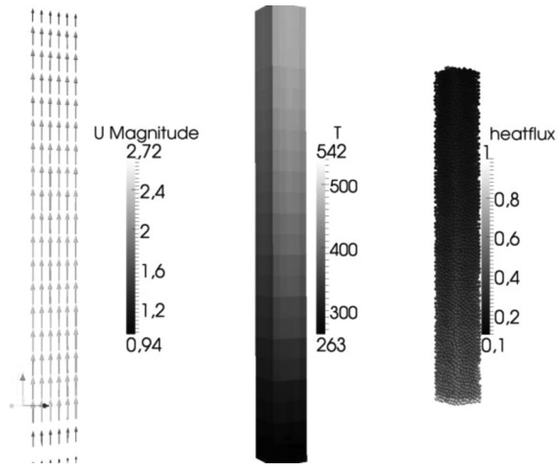


Figure 10 Fixed bed, particles at constant temperature (600 K), air enters from below (with 1 m/s at 263 K) and is heated up



and compared with an analytical solution (Li and Mason, 2000) as depicted in Figure 11.

4.4 Unresolved CFD-DEM: single spout fluidised bed

This test case was originally investigated by Link et al. (2005) and has been excessively investigated during the recent years, e.g., van Buijtenen et al. (2011). It is, therefore, an ideal candidate for checking the CFD-DEM framework using approach A, described in this paper. For volume fraction calculation the 'divided' approach was used. The fluid particle momentum exchange is calculated applying the drag correlation of Koch and Hill (2001).

Geometry

The geometry of the single spout fluidised bed is depicted in Figure 12, the numerical settings are listed in Table 1.

Figure 11 Transient numerical results for the Nusselt number compared to steady state analytical solution

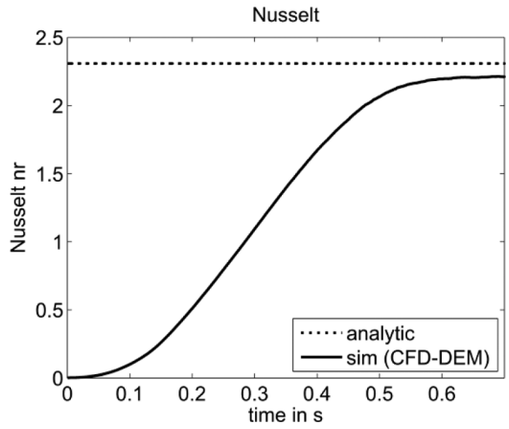


Figure 12 Sketch of the spout fluidised bed geometry

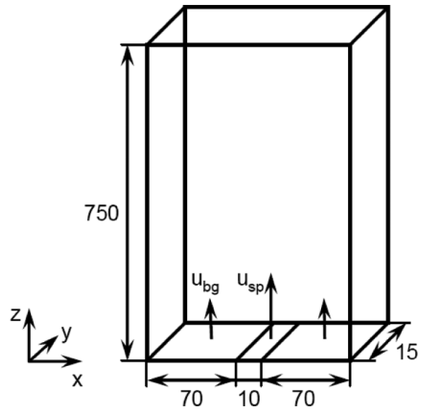


Table 1 Numerical settings

Property	*Value*	*Unit*
N_x	29	–
N_y	2	–
N_z	250	–
t_{end}	20	s
N_p	2.45e4	–

Material properties

The physical and model parameters of the glass beads and the fluid are given in Table 2.

Here, e denotes the coefficient of restitution for particle–particle and particle–wall contact. Accordingly, μ denotes the coefficient of friction.

In Figure 13 the time averaged vertical flux of the granular phase at $z = 0.13$ m is depicted. The results obtained with the model presented in this paper are in very good accordance to data published by Link et al. (2005). Both simulation results slightly differ from experimental data.

4.5 Resolved CFD-DEM: settling of two disks

In this test case the detailed motion and the flow around of two settling disks is calculated using the CFD-DEM framework (approach B). In literature (Patankar et al. (2000)) three states have been defined, namely drafting, kissing and tumbling. The computational domain is a 2D box with 2×6 cm and fixed wall boundary conditions. Material properties and particle starting positions $\mathbf{x}_p$ are set according to Table 3.

In Figure 14 this behaviour and the fluid flow is illustrated. Figures 15 and 16 show the position of the two particles (y-direction) as well as the settling velocities. First, the two particles settle without influencing each other. As soon as the following particle reaches the wake of the leading particle, it accelerates (drafting). At $t \sim 0.15$ s,

Table 2 Material properties/boundary conditions

Property	*Value*	*Unit*
material	glass/air	–
d_p	3.0	mm
ρ_f	1	kg/m^3
ρ_p	2505	kg/m^3
ν_f	1.8e-5	m^2/s
e_{p-p}	0.97	–
e_{p-w}	0.97	–
μ_{p-p}	0.1	–
μ_{p-w}	0.1	–
u_{bg}	1.5	m/s
u_{sp}	30	m/s

Figure 13 Vertical time averaged flux at $z = 0.13$. Results compared to simulation and experimental data achieved by Link et al. (2005)

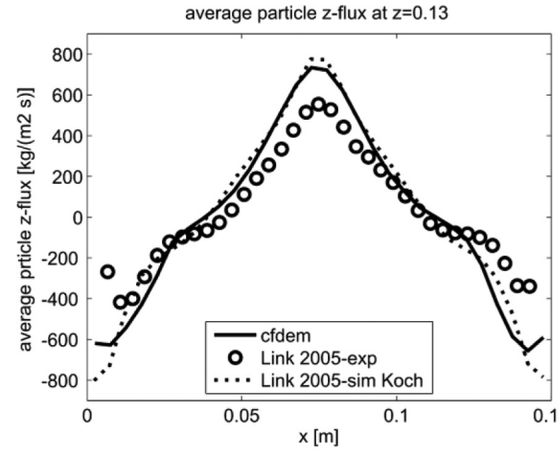


Table 3 Material properties

Property	*Value*	*Unit*
d_p	0.25	cm
ρ_p	1.5	g/cm^3
ρ_f	1	g/cm^3
ν_f	0.1	cm^2/s
$\mathbf{x}_{\mathbf{p1}}$	(1, 4.5)	cm
$\mathbf{x}_{\mathbf{p2}}$	(1, 5)	cm
$\mathbf{g}$	(0, 981)	cm^2/s

Figure 14 Drafting (left), kissing (middle), tumbling (right) and appellation according to Patankar et al. (2000)

Figure 15 Simulated *y*-position of two settling discs, compared to data from Glowinski et al. (2000)

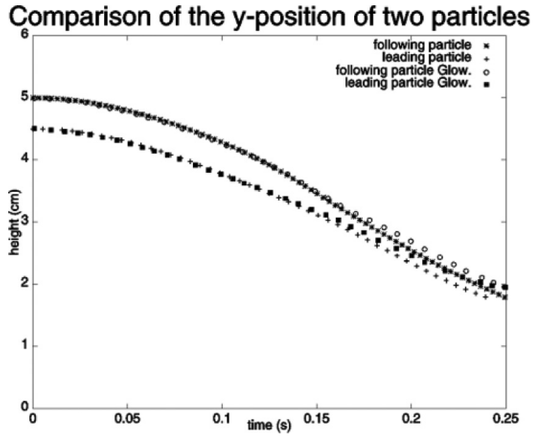
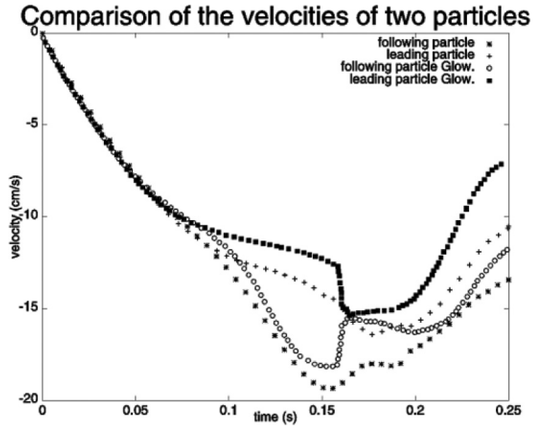


Figure 16 Simulated *y*-velocity of two settling discs, compared to data from Glowinski et al. (2000)



the two particles collide (kissing). Until this point the behaviour of the particles is predictable, only in the last stadium, the so called tumbling, a strict prediction is not possible.

5 Conclusion

Concluding, we show the versatility and applicability of the open source CFD-DEM framework (CFDEM, 2011), which is based on the DEM code LIGGGHTS (LIGGGHTS, 2011) and the open source CFD toolbox OpenFOAM® (OpenCFD Ltd., 2009). LIGGGHTS and the CFD-DEM coupling are available for public download. With 'approach A' we show the applicability of the coupling framework for flows, where the particle sizes are significantly smaller than the CFD grid. The motion of an incompressible fluid phase in the presence of a secondary particulate phase is then governed by a modified set of Navier–Stokes equations accounting for the volume fraction occupied by the fluid, and a momentum exchange term.

Second, with 'approach B' we show the applicability of the coupling framework to the case of large particles and fine computational grids using the fictitious domain/Immersed Boundary (IB) method. In the first step the incompressible Navier–Stokes equations are solved over the whole domain. The next task is to correct the bodies' velocities in the affected cells (i.e., those cells, which are covered by the immersed bodies). Finally, a correction–operation is applied to account for the divergence-free condition of the flow field. Both approaches are successfully tested against analytics as well as experimental data. Validation examples of the coupling are shown, ranging from to dry hopper flow and heat conduction in a packed bed to convective fluid–particle heat transfer and fluidised bed (approach A) to sedimentation (approach B).

Statement

This paper is based on the following two papers from Proc. of 8th International Conference on CFD in the Oil and Gas, Metallurgical and Process Industries (CFD, 2011), Trondheim, Norway 21_23, June 2011:

Kloss, C., Goniva, C., Amberger S. and Pirker, S. (2011) 'Liggghts open source dem: models, features, parallelism and quality assurance',

Goniva, C., Kloss, C., Hager, A., Wierink, G. and Pirker, S. (2011) 'A multi-purpose open source CFD-DEM approach',

All Sections in this paper except for Sections 2.1.3, 2.2.2, 4.2 and 4.3 are based on these two papers. Sections 2.1.3, 2.2.2, 4.2 and 4.3 have been added.

References

CFDEM (2011) '*CFDEM – Open Source CFD, DEM and CFD*, URL http://www.cfdem.com

Chapman, S. and Crowling, T.G. (1970) *The Mathematical Theory of Non-Uniform Gases*, Cambridge University Press, Cambridge.

Chaudhuri, B., Muzzio, F.J. and Tomassone, M.S. (2006) 'Modelling of heat transfer in granular flow in rotating vessels', *Chem. Eng. Sci.*, Vol. 61, pp.6348–6360.

Cundall, P.A. and Strack, O.D. (1979) 'A discrete numerical model for granular assemblies', *Geotechnique*, Vol. 21, pp.47–65.

di Renzo, A. and di Maio, F.P. (2004) 'Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes', *Chemical Engineering Science*, Vol. 59, pp.525–541.

Ergun, S. (1952) 'Fluid flow through packed columns', *Chem. Eng. Prog.*, Vol. 48, No. 2, p.89.

Gidaspow, D., Bezburuah, R. and Ding, J. (1992) 'Hydrodynamics of circulating fluidized beds, kinetic theory approach', in Fluidization VII, *Proceedings of the 7*th *Engineering Foundation Conference on Fluidization*, Brisbane, Australia.

Glowinski, R., Pan, T-W., Hesla, T.I., Joseph, D.D. and Periaux, J. (2000) 'A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: applicatioin to particulate flow', *J. Comput. Phys.*, Vol. 169, pp.363–426.

Hendrickson, B. and Devine, K. (2000) 'Dynamics load balancing in computational mechanics', *J. Comp. Meth. Appl. Engnrg.*, Vol. 184, pp.485–500.

Issa, R. (1986) 'Solution of the implicitly discretised fluid flow equations by operator-splitting', *Journal of Computational Physics*, Vol. 62, No. 1, pp.40–65.

Kafui, K.D., Thornton, C. and Adams, M.J. (2002) 'Discrete particle-continuum modelling of gas–solid fluidised beds', *Chemical Engineering Science*, Vol. 57, pp.2395–2410.

Kloss, C., Goniva, C. and Pirker, S. (2010) 'LIGGGHTS – a new open source dem simulation software', *Proc. 5th Intl. Conf. on Discrete Element Methods*, August, London, pp.25–26.

Kloss, C., Goniva, C., Aichinger, G. and Pirker, S. (2009) 'Comprehensive DEM-DPM-CFD simulations: model synthesis, experimental validation and scalability', *Proceedings Seventh International Conference on CFD in the Minerals and Process Industries*, 9–11 December, CSIRO, Melbourne, Australia.

Koch, D.L. and Hill, R.J. (2001) 'Inertial effects in suspension and porous-media flows', *Annual Review of Fluid Mechanics*, Vol. 33, p.619.

Li, J. and Mason, D.J. (2000) 'A computational investigation of transient heat transfer in pneumatic transport of granular particles', *Powder Technology*, Vol. 112, No. 3, pp.273–282, ISSN 0032-5910.

LIGGGHTS (2011) *LAMMPS Improved for General Granular and Granular Heat Transfer Simulations*, URL http://www.liggghts.com

Link, J.M., Cuypers, L.A., Deen, N.G. and Kuipers, J.A.M. (2005) 'Flow regimes in a spout-fluid bed: a combined experimental and simulation study', *Chemical Engineering Science*, Vol. 60, No. 13, pp.3425–3442, ISSN 0009-2509.

Ogawa, S. (1978) 'Multitemperature theory of granular materials', *Proc. US-Japan Siminar on Continuum-Mechanical and Statistical Approaches in the Mechanics of Granular Materials*, Gakujutsu bunken Fukukai, Tokyo, p.208.

OpenCFD Ltd. (2009) *OpenFOAM – The Open Source CFD Toolbox*, URL http://www.openfoam.com

Patankar, N.A., Singh, P., Joseph, D.D., Glowinski, R. and Pan, T-W. (2000) 'A new formulation of the distributed lagrange multiplier/fictitious domain method for particulate flows', *Int. J. Multiphase Flow*, Vol. 26, pp.1509–1524.

Plimpton, S.J. (1995) 'Fast parallel algorithms for short-range molecular dynamics', *J. Comp. Phys.*, Vol. 117, pp.1–19 (LAMMPS homepage: http://lammps.sandia.gov).

Pöschel, T. and Schwager, T. (2005) *Computational Granular Dynamics*, Springer, Berlin.

Rao, K. and Nott, P. (2008) *An Introduction to Granular Flow*, Cambridge Series in Chemical Engineering, Cambridge.

Sandia (2011) *Online Resource*, http://lammps.sandia.gov/bench.html#chute, accessed on 20 October 2011.

Shirgaonkar, A.A., Maciver, M.A. and Patankar, N.A. (2008) 'A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion', *J. Comput. Phys.*, Vol. 160, pp.2366–2390.

Tsuji, T., Yabumoto, K. and Tanaka, T. (2008) 'Spontaneous structures in three-dimensional bubbling gas-fluidized bed by parallel DEM–CFD coupling simulation', *Powder Technology*, Vol. 184, pp.132–140.

van Buijtenen, M.S., van Dijk, W.J., Deen, N.G., Kuipers, J.A.M., Leadbeater, T. and Parker, D.J. (2011) 'Numerical and experimental study on multiple-spout fluidized beds', *Chemical Engineering Science*, doi:10.1016/j.ces.2011.02.055.

Verlet, L. (1967) 'Computer 'experiments' on classical fluids: I. thermodynamical properties of lennard-jones molecules', *Phys. Rev.*, Vol. 159, pp.98–103.

Wen, C-Y. and Yu, Y.H. (1966) 'Mechanics of fluidization', *Chem. Eng. Prog. Symp. Series*, Vol. 62, p.100.

Zhu, H.P., Zhou, Z.Y., Yang, R.Y. and Yu, A.B. (2007) 'Discrete particle simulation of particulate systems: theoretical developments', *Chemical Engineering Science*, Vol. 62, No. 13, July, pp.3378–3396.

Zhu, H.P., Zhou, Z.Y., Yang, R.Y. and Yu, A.B. (2008) 'Discrete particle simulation of particulate systems: a review of major applications and findings', *Chemical Engineering Science*, Vol. 63, pp.5728–5770.

Bibliography

Bertrand, F., Leclaire, L-A. and Levecque, G. (2005) 'DEM-based models for the mixing of granular materials', *Chemical Engineering Science*, Vol. 60, pp.2517–2531.

Campbell, C.S. (1990) 'Rapid granular flows', *Annual Rev. Fluid Mech.*, Vol. 22, pp.57–92.

Dobby, G.S. and Finch, J.A. (1987) 'Particle size dependence in flotation derived from a fundamental model of the capture process', *Int. J. Miner. Process*, Vol. 21, pp.241–260.

Enwald, H., Peirano, E. and Almstedt, E. (1996) 'Eulerian two-phase flow theory applied to fluidization', *Int. J. Multiphase Flow*, Vol. 22, pp.21–66.

Goniva, C., Kloss, C. and Pirker, S. (2009) 'Towards fast parallel CFD-DEM: an open-source perspective', *Proc. of Open Source CFD International Conference*, 12–13 November, Barcelona.

Goniva, C., Kloss, C., Hager, A. and Pirker, S. (2010) 'An Open Source CFD-DEM Perspective, *Proc. of OpenFOAM Workshop*, 22–24 June, Gothenburg.

Grosse, J., Lange, C., Burow, K., Goniva, C., Kloss, C. and van Zoest, T. (2011) 'First design concept for a combined

thermal and mechanical penetration device for investigations of icy planetary bodies – the cryo-mole', *European Geosciences Union*, Vienna, Austria.

Kloss, C., Goniva, C. and Pirker, S. (2010) 'Open source DEM and CFD-DEM with LIGGGHTS and OpenFOAM®, *Proc. Open Source CFD International Conference*, 4–5 November, Munich.

Kremmer, M. and Favier, J.F. (2001) 'A method for representing boundaries in discrete element modelling – part I: geometry and contact detection', *Int. J. Num. Meth. Eng.*, Vol. 51, No. 12, pp.1407–1421.

Kunii, D. and Levenspiel, O. (1992) *Fluidization Engineering*, 2nd ed., Butterworth-Heinemann Series in Chemical Engineering, ISBN 0-409-90233-0.

Matuttis, H.G., Luding, S. and Herrmann, H.J. (2000) 'Discrete element simulations of dense packings and heaps made of spherical and non-spherical particles', *Powder Technology*, Vol. 109, pp.278–292.

SANDIA (2009) *LAMMPS User Manual*, http://lammps.sandia.gov/doc/Manual.html, Sandia National Laboratories, USA.

Schulze, H.J. (1983) 'Physico-chemical elementary processes in flotation', *Vol. 4 of Developments in Mineral Processing*, Elsevier, Amsterdam, The Netherlands.

Staknis, M. (1990) 'Software quality assurance through prototyping and automated testing', *J. Information and Software Technology*, Vol. 32, No. 1, pp.26–33.

Wu, C.L. and Berrouk, A.S. (2010) 'Conceptual comparative study between efficient collision-handling algorithms used for Molecular Dynamics (MD) and Discrete Particle Model (DPM)', *Powder Technology*, Vol. 198, pp.435–438.

Zhou, Y.C., Wright, B.D., Yang, R.Y., Xu, B.H. and Yu, A.B. (1999) 'Rolling friction in the dynamic simulation of sandpile formation', *Physica. A*, Vol. 269, pp.536–553.

Nomenclature

Greek symbols	
α	Volume fraction [1]
$\boldsymbol{\Gamma}$	Boundary
$\boldsymbol{\delta}$	Particle–particle overlap [m]
Φ	Scalar field
κ	Thermal diffusivity [m²/s]
v	Kinematic viscosity [Pa s]
μ_c	Coulomb friction coefficient [1]
ρ	Density [kg/m³]
λ	Thermal conductivity [W/(m K)]
$\boldsymbol{\tau}$	Stress tensor [Pa]
$\boldsymbol{\omega}$	Rotational velocity [1/s]
Ω	Domain
Latin symbols	
A	Particle–particle contact area [m²]
c	Damping coefficient [kg/s]
C_d	Drag coefficient [1]
C	Specific heat capacity [J/(kg K)]
d	Diameter [m]
$\mathbf{F}$	Force [N]
$\mathbf{g}$	Gravity constant [m/s²]
h	Heat transfer coefficient [W/(m² K)]
I	Moment of inertia [kg m²]
k	Stiffness [N/m]
$\mathbf{K}$	Momentum exchange coefficient [ks/(s m²)]
m	Mass [kg]
$\hat{\mathbf{n}}$	Outer normal vector
N	Number of grid cells [1]
Nu	Nusselt number [1]
p	Pressure [Pa]
q	Heat flux [W/m²]
Pr	Prandtl number [1]
q	Heat transfer rate [W]
$\mathbf{R}$	Momentum source term [kg/(m² s²)]
r	Radius [m]
$\mathbf{u}$	Velocity [m/s]
$\Delta\mathbf{u}$	Relative velocity [m/s]
Re	Reynolds number [1]
s	Verlet (skin) parameter [m]
t	Time [s]
Δt	Time-step size [s]
$\mathbf{T}$	Torque [Nm]
T	Temperature [K]
V	Volume [m³]
$\mathbf{x}$	Position [m]
$\Delta\mathbf{x}$	Particle overlap at contact point [m]
Sub/superscripts	
c	Contact
d	Drag
f	Fluid
i	Particle index
n	Normal to contact point
p	Particle
r	Rolling friction
s	Solid
t	Tangential to contact point