Eleventh International Conference on CFD in the Minerals and Process Industries
CSIRO, Melbourne, Australia
7-9 December 2015

OPEN SOURCE CFD-DEM MODELLING FOR PARTICLE-BASED PROCESSES

Christoph GONIVA[1*], Bruno BLAIS[2], Stefan RADL[3], Christoph KLOSS[1]

[1] DCS Computing GmbH, Linz, AUSTRIA

[2] Chemical Engineering Departement, Ecole Polytechnique de Montreal, CANADA

[3] Inst. f. Process and Particle Engineering, Graz University of Technology, AUSTRIA

*Corresponding author, E-mail address: office@dcs-computing.com

ABSTRACT

Discrete Element Method (DEM) coupled to Computational Fluid Dynamics (CFD-DEM) is a powerful tool for optimization and design of particle processes. We give an overview of recent developments in the frame of the open source CFD-DEM code CFDEMcoupling (2015) and the open source DEM code LIGGGHTS (2015). In particular, we summarize recent implementations of model strategies for resolved CFD-DEM method using immersed boundary (IB) techniques for handling solid body motion (e.g. rotors) in coupled CFD-DEM simulations. Further we show model improvements in handling interpolation and averaging correctly and avoiding numerical errors. On a more general level, we present a novel parallelization approach and present its capabilities for heavily parallel computations. These model improvement and developments of the CFDEMcoupling (2015) framework must be seen in the context of their industrial applicability (e.g. food processing, agricultural engineering, pharmaceuticals, mining and consumer goods) which is the driving force for any of the afore mentioned developments.

NOMENCLATURE

c	damping coefficient (kg/s)
d	diameter (m)
$\mathbf{f}$	explicit momentum exchange coefficient (N/m^3)
$\mathbf{F}$	force exerted on a single particle (N)
$\mathbf{g}$	gravity vector (m/s2)
$\mathbf{I}$	identity matrix (-)
$\mathbf{K}$	momentum exchange coefficient (kg/(m3 s))
k	spring stiffness (N/m)
m	mass (kg)
N	number (cells, particles) (-)
p	pressure (Pa)
r	radius (m)
$\boldsymbol{R}$	semi-implicit momentum source term (N/m^3)
Re	Reynolds number (-)
$\boldsymbol{T}$	torque (Nm)
t	time (step) (s)
$\mathbf{u}$	velocity (m/s)
$\Delta\boldsymbol{u}_p$	relative particle velocity at contact point (m/s)
V	volume (m^3)
$\boldsymbol{x}$	position (m)
x, y, z	Cartesian-coordinates (-)
$\Delta\boldsymbol{x}$	particle overlap at contact point (m)
α	volume fraction (-)
ν_p	Poisson ratio (-)
λ_f	bulk viscosity (kg/(m s))
μ_c	Coulomb friction coefficient (-)
μ_f	gas phase shear viscosity (kg/(m s))
μ	friction coefficient (-),dynamic viscosity (kg/(m s))
ρ	density (kg/m3)
$\boldsymbol{\tau}$	surface tension (N/m)
$\boldsymbol{\omega}$	angular velocity (1/s)

Sub/superscripts

c	contact
f	fluid
n	normal to contact point
p	particle
t	tangential to contact point
rel	relative

INTRODUCTION

The coupled CFD-DEM approach described in this paper is implemented within an open source environment (CFDEMcoupling, 2011), which has seen a steady development since its kick-off by Goniva (2009) and Kloss (2012). The vision of this development is providing a general and versatile toolbox for modelling fluid-granular systems. Since then an incredible number of researchers and engineers all over the world have been using and improving this code base. The corresponding author's affiliation (DCS Computing GmbH) was founded and gives a professional base for further developments and improvements.

In this paper we address some of the latest improvements and development in the CFDEMcoupling framework. For sake of completeness we first provide a brief model description of the unresolved CFD-DEM method. Then we describe a novel approach for moving and rotating objects within a CFD simulation using a forcing immersed boundary method. We show a validation of this method for low Reynolds number flows as well as its combination with the unresolved CFD-DEM method.

Following the idea of implicit-explicit force splitting presented recently by Radl et al. (2015) we highlight numerical details for calculating momentum exchange terms and provide solutions for correcting interpolation and averaging errors.

In a final chapter we briefly show some novel developments on efficient parallel data exchange which highly influences parallel scalability.

Copyright © 2015 CSIRO Australia

MODEL DESCRIPTION

DEM Method

The Discrete Element Method 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 (e.g. Di Renzo & Di Maio, 2004).

The strength of the DEM lies in its ability to resolve the granular medium at the particle scale, thus allowing for 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, the DEM is able to capture many different physical phenomena, such as dense and dilute particulate regimes, rapid- as well as slow granular 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 actual 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. LAMMPS, 2009).

Governing Equations

In the framework of the DEM, all particles in the computational domain are tracked in a Lagrangian way, explicitly solving each particle's trajectory, based on the force and torque balances:

$$\mathbf{m}_p\,\ddot{\mathbf{x}}_p = \mathbf{F}_{p,n} + \mathbf{F}_{p,t} + \mathbf{F}_{p,f} + \mathbf{F}_{p,p} + \mathbf{F}_{p,v} + \mathbf{F}_{p,b} \quad (1)$$

and

$$\mathbf{I}_\mathbf{p}\,\frac{d\boldsymbol{\omega}_\mathbf{p}}{dt} = \mathbf{r}_{p,\mathbf{c}} \times \mathbf{F}_{p,t} + \mathbf{T}_{p,r} \quad (2)$$

where $\mathbf{F}_{p,n}$ is the normal contact force, $\mathbf{F}_{p,t}$ is the tangential contact force. $\mathbf{F}_{p,f}$ is the drag force exerted from the fluid phase to the particles, $\mathbf{F}_{p,p}$ and $\mathbf{F}_{p,v}$ denote respectively the pressure and viscous force acting on the particles. Other body forces like gravity, electrostatic or magnetic forces are lumped into $\mathbf{F}_{p,b}$ For sake of completeness, these forces are described in detail in Table 1.

Each physical 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. Generally, the particles are allowed to overlap slightly. The normal force tending to repulse the particles can then be deduced from this spatial overlap $\Delta\boldsymbol{x}_p$ and the normal relative velocity at the contact point, $\Delta\boldsymbol{u}_{p,n}$. The simplest example is a linear spring-dashpot model, shown in Fig. 1.

An efficient way of taking into account the small-scale non-sphericity of the particles is a rolling friction model, see Goniva et al. (2012). It introduces an additional torque also for collisions, where the relative velocity at the contact point is zero. Within this paper a directional constant torque model (Ai et al., 2011), is applied (see Tab. 1).

Forces	Correlations
$\mathbf{F}_{p,n}$	$-k_n \Delta\mathbf{x}_p + c_n\,\Delta\mathbf{u}_{p,n}$
$\mathbf{F}_{p,t}$	$\min\left\{ \left\| \mathrm{k_t} \int_{\mathrm{t_{c,0}}}^{\mathrm{t}} \Delta \mathrm{u_{p,t}}\,\mathrm{dt} + \mathrm{c_t}\Delta \mathrm{u_{p,t}} \right\|,\ \mu_\mathrm{c}\,\mathrm{F_{p,n}} \right\}$
$\mathbf{F}_{p,b}$	$\mathbf{g}\ \mathrm{m_p}$
$\mathbf{T}_{p,r}$	$R_\mu k_n \Delta\mathbf{x}_p \frac{\boldsymbol{\omega}_\mathbf{rel}}{\left\|\boldsymbol{\omega}_\mathbf{rel}\right\|}\frac{d}{2}$
$\boldsymbol{\omega}_{rel}$	$\frac{r_{p,ci}\boldsymbol{\omega}_{p,i} + r_{p,cj}\boldsymbol{\omega}_{p,j}}{r_{p,ci} + r_{p,cj}}$
$\mathbf{F}_{p,p}$	$-\nabla(p)V_p$
$\mathbf{F}_{p,v}$	$-\nabla\cdot(\boldsymbol{\tau})V_p$
$\mathbf{F}_{p,f}$	$\mathbf{F}_D$*

*Drag force $\mathbf{F}_D$ is explained in Goniva et al. (2012)

Table 1: Components of forces and torques acting on particle p.

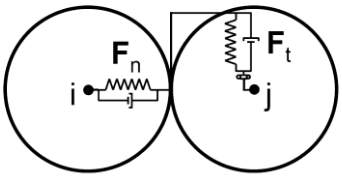

Figure 1 Spring-dashpot model.

CFD-DEM Method

For the modelling of particle laden fluid flow, especially for dense particle flow close to packing limit a coupled CFD-DEM approach (Tsuji et al. (1993), Zhou et al. (2010)) is well suited.

Governing Equations

The motion of a fluid phase in the presence of a secondary particulate phase is governed by the volume-averaged Navier-Stokes Equations for compressible fluid, which can be written as:

$$\frac{\partial(\rho_f\alpha_f)}{\partial t} + \nabla\cdot(\rho_f\alpha_f\,\mathbf{u}_f) = 0, \quad (3)$$

$$\begin{aligned}&\frac{\partial(\rho_f\alpha_f\,\mathbf{u}_f)}{\partial t} + \nabla\cdot(\rho_f\alpha_f\,\mathbf{u}_f\,\mathbf{u}_f) = \\ &-\alpha_f\nabla p + \mathbf{R}_{f,p} + \nabla\cdot(\alpha_f\boldsymbol{\tau}_f) + \mathbf{f}\end{aligned}. \quad (4)$$

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

For solving the above equations a pressure based solver using "Pressure-Implicit Split-Operator" (PISO) pressure-velocity coupling is used, where an implicit momentum predictor followed by a series of pressure solutions and explicit velocity corrections is performed (Jasak, 1996).

Copyright © 2015 CSIRO Australia

CFD-DEM coupling routine

The coupling routine consists of several steps:

(1) The DEM solver calculates the particle positions and velocities.

(2) The particle positions, velocities, and other necessary data are passed to the CFD solver.

(3) For each particle, the corresponding cell in the CFD mesh is determined.

(4) For each cell, the particle volume fraction as well as a mean particle velocity is determined.

(5) Based on the particle volume fraction, the fluid forces acting on each particle are calculated.

(6) Particle-fluid momentum exchange terms are assembled from particle based forces by ensemble averaging over all particles in a CFD cell.

(7) The fluid forces acting on each particle are sent to the DEM solver and used within the next time step.

(8) The CFD solver calculates the fluid velocity taking into account local particle volume fraction and momentum exchange of the granular phase.

(9) Additional equations such as species concentration can optionally be evaluated.

(10) The routine is repeated from (1).

Fluid-Particle momentum exchange

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 local particle volume fraction.

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 $\langle \mathbf{u}_p \rangle$:

$$\mathbf{R}_{f,p} = \mathbf{K}_{f,p}\mathbf{u}_f - \mathbf{K}_{f,p}\left\langle \mathbf{u}_p \right\rangle, \qquad (5)$$

where

$$\mathbf{K}_{f,p} = -\frac{\left| \sum_i \mathbf{F}_{p,f} \right|}{V_{cell} \cdot \left| \mathbf{u}_f - \left\langle \mathbf{u}_p \right\rangle \right|}, \qquad (6)$$

For the calculation of $\mathbf{K}_{f,p}$ many different drag correlations have been proposed during the recent years (e.g. Zhu et al. (2007)). Within this paper a drag relation based on lattice Boltzmann simulations proposed by Koch and Hill (2001) is used, see Goniva et al. (2012).

Coupled CFD-DEM solver

The CFD part of the simulations is realized by a solver realised using the open source framework of OpenFOAM® (OpenFOAM, 2015). The coupling routines are collected in a separate library of > 10k LOC providing a modular framework for CFD-DEM coupling. The DEM part of the simulations is conducted in LIGGGHTS, an open source software package for modelling granular material by means of the Discrete Element Method (LIGGGHTS, 2015). based on LAMMPS, an open source Molecular Dynamics code by Sandia National Laboratories for massively parallel computing on distributed memory machines (Plimpton, 1995).

Both LIGGGHTS and CFDEMcoupling run in parallel using message-passing techniques (MPI) and a spatial-decomposition of the simulation domain. LIGGGHTS and CFDEMcoupling are distributed as open source codes under the terms of the GNU General Public License (GPL). A selection of coupling routines as well as example solvers are provided at a dedicated web page maintained by DCS Computing GmbH.

Novel Immersed Boundary CFD Method

Recently described by Hager et al. (2014) an immersed boundary method has been developed within the CFDEMcoupling framework ("cfdemSolverIB") following the approach of Shirgaonkar et al. (2009). The particle dynamics is modelled using the DEM method, the particle positions, velocities and sizes are transferred to the CFD solver which then uses an immersed boundary method to calculate the flow field around the particles. While this method has proven its capability to predict flows of moderate Reynolds number, it fails for low Reynolds number flows.

Very recently an improved immersed boundary method has therefore been developed by Blais et al. (2015) and implemented in the CFDEMcoupling framework ("cfdemSolverForceIB"). This novel scheme makes use of the intrinsic cycling within the PISO loop to iterate on the continuous forcing term added to the momentum equation to take into account the immersed body and its motion. Blais et al. could show second order convergence of the method in case of a "body-conformal" mesh. For a non-conformal mesh the order of convergence reduces to 1.33.

Immersed Boundary Validation Case

Within this publication we compare the performance of the old and new immersed boundary implementation for low Reynolds number flows. Therefore we consider the flow around a sphere at a Reynolds number of 1. The sphere is positioned in the centre a cubic simulation domain of 10x10x10 diameters, meshed with 20x20x20 cells. The simulation result is validated against analytic solution of the flow past a sphere for Stokes flow:

$$\begin{aligned}\mathbf{u}(\mathbf{r}) = \mathbf{u}_\infty - \frac{3R}{4r}\left(\mathbf{u}_\infty + (\mathbf{e}_r \cdot \mathbf{u}_\infty)\mathbf{e}_r\right) \\ - \frac{R^3}{4r^3}\left(\mathbf{u}_\infty - 3(\mathbf{e}_r \cdot \mathbf{u}_\infty)\mathbf{e}_r\right)\end{aligned}, \qquad (7)$$

where R is the sphere radius, $\mathbf{u}_\infty$ is the far field velocity vector, r is the radial coordinate and **e** the unit vector of the coordinate system as show in Fig. 1.

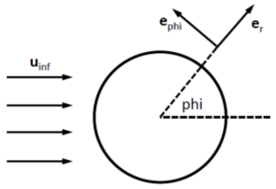

Figure 1: Stokes flow around sphere.

In order to increase the spatial mesh resolution at the surface of the sphere a dynamic meshing routine was applied leading to a maximum refinement factor of 3 at the sphere surface. In Fig. 2 the velocity profiles in span-wise direction (i.e. phi=90°) of flow past a sphere at Re=1 are depicted. The simulation results for "cfdemSolverIB" and "cfdemSolverForceIB" are compared to the analytical solution for Stokes flow (Eqn. 7) and the analytical solution for Potential flow around a sphere. It can clearly be seen that the new forcing based immersed boundary method is able to nicely predict the flow close the sphere surface. The old method fails to predict zero velocity at the sphere position and predicts a solution similar to potential flow solution.

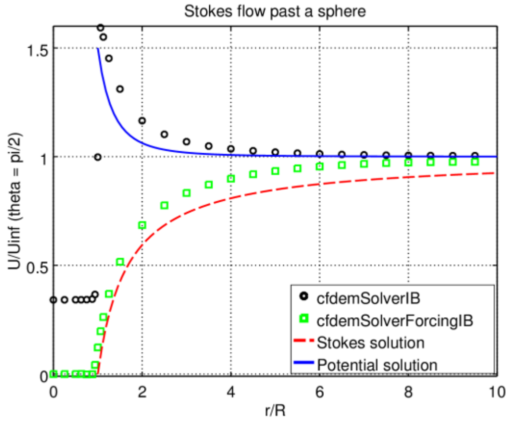

Figure 2: Flow profile in span-wise direction of flow past a sphere at Re=1. Comparison of “cfdemSolverIB” (Hager et. Al. 2014) and “cfdemSolverForceIB” (Blais et al. 2015) with the analytic solution for Stokes flow.

In Fig. 3 the contour of the velocity magnitude ranging from 0 to 1 $\mathbf{u}_\infty$,which is in positive x direction, are shown.

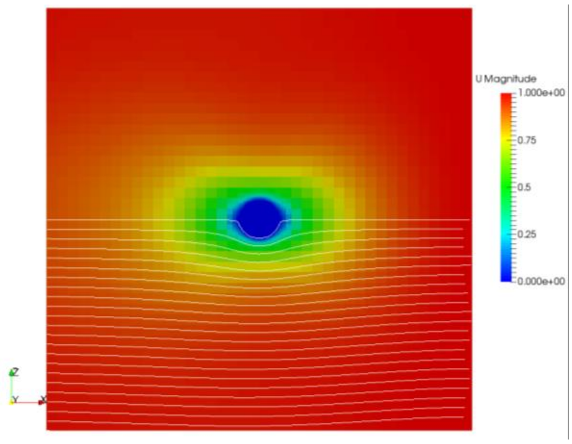

Figure 3: Contour of the velocity magnitude (m/s) in a slice through the sphere, flow from left to right in positive x-direction.

Immersed Boundary CFD-DEM Method

Combining the forcing immersed boundary method and unresolved CFD-DEM method we created a model for complex rigid body motion (e.g. stirrers) in a particle-fluid suspension. As described in more detail by Blais et al. (2015), rotating and moving solid parts are resolved by the forcing immersed boundary method in the CFD domain as well as by moving surfaces in the DEM domain. Particles are modelled with classical unresolved CFD-DEM method. Compared to modelling rotating and moving parts by mesh motion or mesh deformation this new method can model multiples solid objects for which the swept volumes overlap.

Immersed Boundary CFD-DEM Validation Case

As a validation case of this solver we present compare the particle motion in a Taylor-Couette flow at Reynolds number 0.7. The setup consists of a rotating inner cylinder and a fixed outer cylinder depicted in Fig. 4. The geometrical scales and boundary conditions are listed in Tab. 2.

Property	Value
$\mathbf{d}_{inner}$	0.0128 m
$\mathbf{d}_{outer}$	0.0476 m
$\mathbf{d}_{Particle}$	0.001 m
$\boldsymbol{\nu}$	0.001 m^2/s
$\boldsymbol{\rho}$	1000 kg/m^3
$\boldsymbol{\rho}_{Particle}$	2000 kg/m^3
$\boldsymbol{\Omega}$	60 rpm
$\mathbf{n}_{Cells}$	2000 (pseudo 2D)

Table 2: Boundary conditions for Taylor-Couette case.

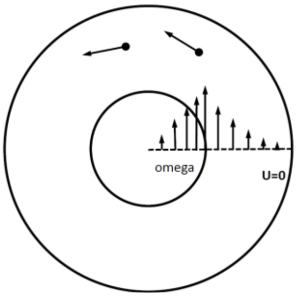

Figure 4: Sketch of geometry setup of Taylor-Couette flow.

Due to the low particle Stokes number it can be assumed that the particle has to follow the fluid flow exactly. As a consequence we can compare the analytic solution for the fluid flow and the particle motion. In Fig. 5 we see that the particle and flow velocity compare very well, which leads to the conclusion that the forcing immersed boundary method correctly predicts the fluid flow and that the particle trajectory calculation is stable and accurate enough. Especially the fact that the cells are not aligned with the flow (Fig. 6) direction is demanding for the numerical accuracy.

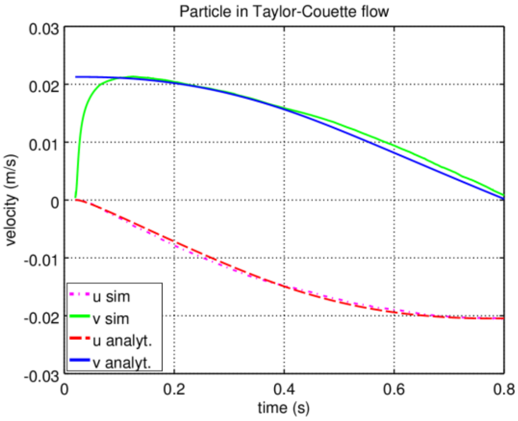

Figure 5: Velocity components of a fine particle in Taylor-Couette flow at Re=0.885 compared to the analytical flow solution.

Copyright © 2015 CSIRO Australia

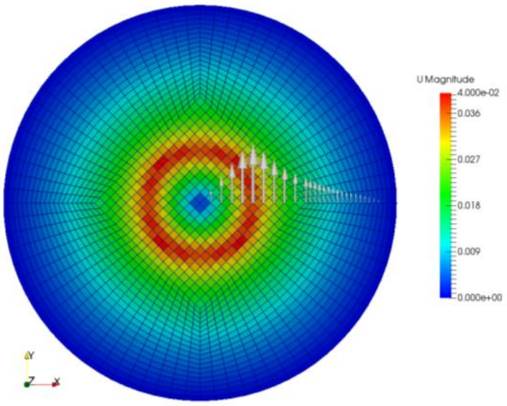

Figure 6: Velocity contour (m/s) and mesh resolution of the Taylor-Couette setup.

Correction Numerical Errors in Unresolved CFD-DEM

Correction of Particle Velocity Averaging Error

When calculating the momentum exchange between particle and fluid phase for unresolved CFD-DEM, the Eqs. 5 and 6 are solved. **F**$_{,f,p}$ is the drag force of each single particle:

$$\mathbf{F}_{p,f} = \frac{V_p\beta}{\alpha_p}\left(\mathbf{u}_f - \mathbf{u}_p\right) \tag{8}$$

, where β is a force coefficient according to a suitable drag calculation (see Zhu et al. (2007)). The interested reader now sees that in the Eqs. 5, 6 and 8 there is a discrepancy between the use of relative particle velocity $\left(\mathbf{u}_f - \mathbf{u}_p\right)$ and the ensemble averaged relative particle velocity $\left(\mathbf{u}_f - \langle\mathbf{u}_p\rangle\right)$. This discrepancy causes an error and leads to reduced stability of the simulation. In order to correct that we can define the error for each particle:

$$\mathbf{u}'_p = \left(\mathbf{u}_p - \langle\mathbf{u}_p\rangle\right) \tag{9}$$

As recently described by Radl et al. (2015) the force-splitting into implicit and explicit part can dramatically increase the numerical stability of coupled CFD-DEM simulations. In this paper we use the force-splitting for error correction:

The semi-implicit contribution of the drag force is calculated as

$$\frac{V_p\beta}{\alpha_p}\left(\mathbf{u}_f - \mathbf{u}_p + \mathbf{u}'_p\right) \tag{10}$$

And an explicit error correction is applied

$$-\frac{V_p\beta}{\alpha_p}\mathbf{u}'_p \tag{11}$$

As a consequence, **K**$_{f,p}$ is formed completely using $\left(\mathbf{u}_f - \langle\mathbf{u}_p\rangle\right)$ and the sum of the explicit and semi-implicit source term give the desired drag force $\frac{V_p\beta}{\alpha_p}\left(\mathbf{u}_f - \mathbf{u}_p\right)$.

Correction of Fluid Velocity Interpolation Error

Without doubt it is desired to use the fluid velocity interpolated to the particle centre $\mathbf{u}_{f,\mathrm{int}}$ when calculating the particle drag force (Eqn. 8), in order to avoid unphysical formation of clusters and shear bands at cell borders. Eqn. 8 can then be re-written as $\mathbf{F}_{p,f} = \frac{V_p\beta}{\alpha_p}\left(\mathbf{u}_{f,\mathrm{int}} - \mathbf{u}_p\right)$.

Similar to the previously described particle velocity averaging error, the discrepancy in using $\mathbf{u}_{f,\mathrm{int}}$ for particle force calculation and the cell centred fluid velocity $\mathbf{u}_f$ for the calculation of **K**$_{,f,p}$ leads to an error which needs to be corrected.

We can define an error for each particle:

$$\mathbf{u}'_f = \left(\mathbf{u}_{f,\mathrm{int}} - \mathbf{u}_f\right) \tag{12}$$

We can now calculate the semi-implicit contribution of the drag force as

$$\frac{V_p\beta}{\alpha_p}\left(\mathbf{u}_f - \mathbf{u}_p + \mathbf{u}'_f\right) \tag{13}$$

And an explicit error correction

$$-\frac{V_p\beta}{\alpha_p}\mathbf{u}'_f \tag{14}$$

As a consequence, **K**$_{f,p}$ is formed completely using $\left(\mathbf{u}_f - \mathbf{u}_p\right)$ and the sum of the explicit and semi-implicit source term give the desired drag force $\frac{V_p\beta}{\alpha_p}\left(\mathbf{u}_{f,\mathrm{int}} - \mathbf{u}_p\right)$.

Effect of Interpolation Error Correction

The consequence and necessity of correcting the error induced by fluid velocity interpolation when calculating the drag force is demonstrated for the onset of fluidization of a particle bed. It should be mentioned that for this specific case it is not to be expected that interpolation changes the result significantly as velocity gradients are not dominating the flow behaviour. Any influence of interpolation can therefore be interpreted as numerical error.

In Fig. 7 the test setup is depicted. In a cylindrical tube a fluid flows in in counter-gravity direction through particle bed which is initially at rest. The flow velocity is slowly increased, which leads to an increasing pressure drop over the packing. When the minimum fluidization velocity is reached, the particles start lifting which leads to decreased packing density and the pressure drop remains almost constant.

Copyright © 2015 CSIRO Australia

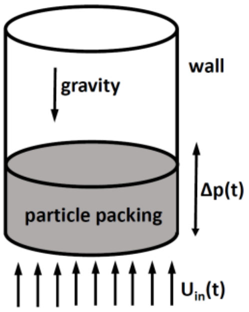

Figure 7: Sketch of geometry setup of minimum fluidization test setup.

In Tab 3 we see a collection of the boundary conditions and properties.

Property	Value
$\mathbf{r}_{tube}$	0.0138 m
$\mathbf{h}_{tube}$	0.0053 m
$\mathbf{d}_{Particle}$	0.001 m
$\boldsymbol{\nu}$	0.00015 m^2/s
$\boldsymbol{\rho}$	10 kg/m^3
$\boldsymbol{\rho}_{Particle}$	2000 kg/m^3
$\mathbf{u}_{\mathbf{in}}$	0-0.02 m/s
$\mathbf{n}_{Cells}$	4608
$\mathbf{n}_{Particles}$	10000

Table 3: Properties and boundary conditions of the minimum fluidization test setup.

In Fig. 8 we see the pressure drop over the particle packing over the inlet velocity in comparison to the analytic solution provided by the well-established and generally applicable Ergun equation and pressure drop at minimum fluidization velocity where particle mass and pressure drop must be balanced. It can be seen that the pressure signal becomes very noisy when using the interpolated fluid velocity $\mathbf{u}_{f,\text{int}}$ for particle drag calculation. This clearly shows the effect of using interpolation for drag force calculation.

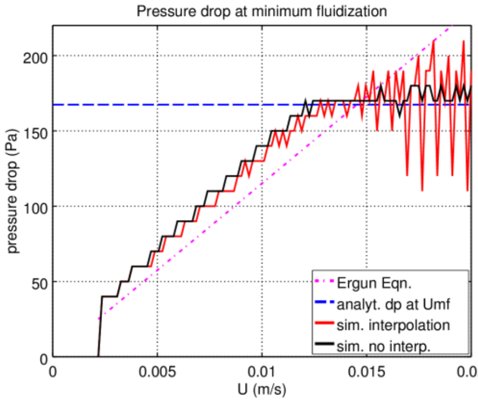

Figure 8: Comparison of pressure drop signal for using either $\mathbf{u}_f$ (black line) or $\mathbf{u}_{f,\text{int}}$ (red line) for particle drag calculation.

In a next step we show how the interpolation error can be corrected efficiently by using the correction velocity $\mathbf{u}'_f$.

Fig. 9 shows that the pressure drop signal with and without interpolation almost match if the correction is applied.

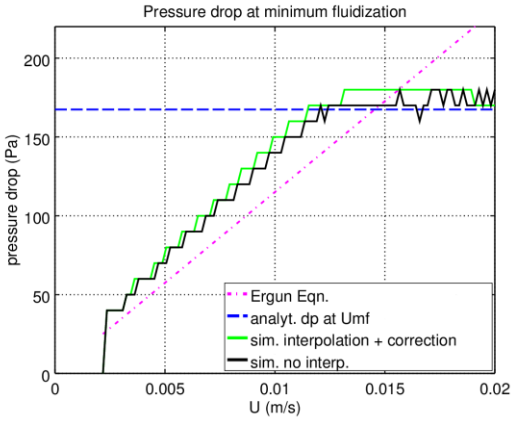

Figure 9: Comparison of pressure drop signal for using either $\mathbf{u}_f$ (black line) or $\mathbf{u}_{f,\text{int}}$ and the error correction (green line) for particle drag calculation.

Efficient Communication Scheme

In coupled CFD-DEM simulations, the communication of data between the CFD solver and the DEM solver is crucial part, especially when using different codes for DEM and CFD. This communication can dominate the simulation time and can become the limiting factor for parallel code scalability for high parallelization (> 64 cores).

In the most general approach is to exchange all particles' data (velocity, position and radius) to the CFD solver and send back all particles' drag forces. This approach is known as "all-to-all" communication. This approach is simple to implement but produces increasing communication overhead with increasing number of cores.

In this paper we present an alternative code-coupling scheme for heterogenious domain decomposition (many-to-many). Equal to the all-to-all scheme it allows for a different domain decomposition for the CFD and the DEM domain of a coupled CFD-DEM simulation and therefore allows for dynamic load balancing of either of the solvers or different domain sizes.

For the many-to-many scheme the single DEM and CFD processes directly communicate and exchange their data. The communication follows a "communication map" which is built and maintained depending on the actual position of each particle.

Testing Parallel Scalability

To test the communication scheme a packed bed consisting of a block 10.24 x 0.002 x 0.1 m and 10240 x 2 x 100 cells was filled with particles of d_P = 0.3 mm. The total particle number was n_P=20.48e6. A schematic description of the setup is shown in Fig. 10.

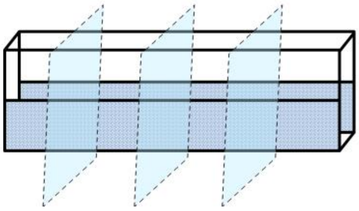

Figure 10: Rectangular packed bed with domain decomposition.

Copyright © 2015 CSIRO Australia

In Fig. 11 the parallel scalability of the system on 8-512 cores is plotted over the number of cores. We see that up to 512 cores the global simulation time decreases over-linearly. The effect of “better than ideal” performance is well known from CFD simulations and is caused by cache demands of large simulations and therefore “not-ideal” performance of low parallelized cases.

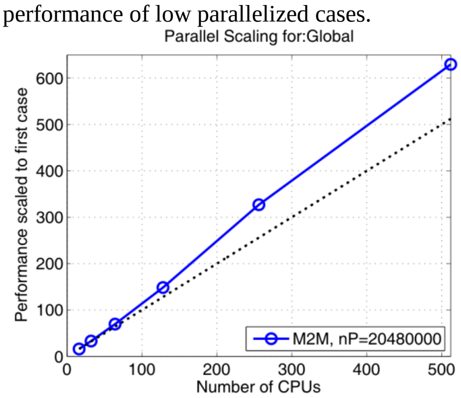

Figure 11: Parallel scalability of the overall simulation time of the new code-coupling scheme.

CONCLUSION

We presented the validation of a novel IB method in the CFDEMcoupling (2015) which allows the combination of IB and CFD-DEM. Further we showed the increase of numerical stability by correction of numerical errors due to interpolation and averaging during particle drag force calculation. Finally we briefly described a novel code communication scheme for CFD-DEM simulations leading to excellent code scalability up to 512 cores.

ACKNOWLEDGEMENTS

The authors would like to thank the community of CFDEMcoupling (2015) for supporting this open source project and the European Commission which funded some parts of the work presented here in the frame of the “NanoSim” Project (Grant Agreement No.: 604656).

REFERENCES

AI, J., CHEN, J.F., ROTTER, J.M., & OOI, J.Y., (2011), “Assessment of rolling resistance models in discrete element simulations”, *Powder Technology*, **206(3)**, 269-282.

BLAIS, B.; LASSAIGNE, M.; GONIVA, C.; FRADETTE, L.; BERTRAND, F. (2015), “A Semi-Implicit Immersed Boundary Method and its Application to Viscous Mixing”, *Computers & Chemical Engineering, submitted for publication*

CFDEMcoupling (2015). *CFDEM — Open source CFD, DEM and CFD*. Retrieved Feb. 26, 2015, from http://www.cfdem.com

CUNDALL, P.A., & STRACK, O.D. (1979). “A discrete numerical model for granular assemblies.” *Geotechnique*, **29**, 47-65.

Di RENZO, A., & Di MAIO, F.P., (2004), „Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes”, *Chemical Engineering Science*, **59**, 525-541.

GONIVA, C., KLOSS, C. and PIRKER, S., (2009), “Towards fast parallel CFD-DEM: An open-source perspective”, In *Proceedings of Open Source CFD International Conference*, Barcelona, Spain.

GONIVA, C., KLOSS, C., DEEN, N.G., KUIPERS, J.A.M., and PIRKER, S., (2012), “Influence of rolling friction on single spout fluidized bed simulation”, *Particuology*, DOI 10.1016/j.partic.2012.05.002.

HAGER, A.; KLOSS, C.; PIRKER, S.; GONIVA, C., (2014), “Parallel Resolved Open Source CFD-DEM: Method, Validation and Application”, *Journal of Computational Multiphase Flows*, **6(1)**, pp. 13-27, DOI: 10.1260/1757-482X.6.1.13

JASAK, H., (1996), “Error analysis and estimation for the finite volume method with applications to fluid flows.” Doctoral dissertation, Imperial College, London.

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*, **12**, No.2/3 , pp.140 – 152.

KOCH, D.L., and HILL, R.J., (2001), “Inertial effects in suspension and porous-media flows”, *Annual Review of Fluid Mechanics*, **33**, 619-647.

LAMMPS (2009). “LAMMPS user manual. Sandia National Laboratories”, USA. Retrieved October 20, 2011, from http://lammps.sandia.gov/doc/Manual.html

LIGGGHTS (2015). “LAMMPS improved for general granular and granular heat transfer simulations.” Retrieved July 21, 2015, from http://www.cfdem.com

OpenFOAM, The OpenFOAM Foundation (2015), “OpenFOAM — The open source CFD toolbox.” Retrieved May 22, 2015, from http://www.openfoam.org

PLIMPTON, S.J., (1995), “Fast parallel algorithms for short-range molecular dynamics”, *Journal of Computational Physics*, **117**, 1-19. (LAMMPS homepage: http://lammps.sandia.gov).

RADL, S., GONZALES, B., GONIVA, C., PIRKER, S. (2015) “STATE OF THE ART IN MAPPING SCHEMES FOR DILUTE AND DENSE EULER-LAGRANGE SIMULATIONS”, Progress in Applied CFD, Sintef Academic Press, pp 103-112, ISBN 978-82-536-1433-5

SHIRGAONKAR, A. A.; MACIVER, M.A.; PATANKAR, A. (2009), “A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion”, *J. Comput. Physics*, **228**, pp. 2366-2390

TSUJI, Y., KAWAGUCHI, T., & TANAKA, T., (1993), “Discrete particle simulation of two-dimensional fluidized bed”, *Powder Technology*, **77**, 79-87.

ZHOU, Z. Y.; KUANG, S. B.; CHU, K. W.; YU, A. B., (2010), “Discrete particle simulation of particle-fluid flow: model formulations and their applicability”, *Journal of Fluid Mechanics*, **661**, pp. 482-51

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, **62**, pp 3378-3396.