PARALLEL OPEN SOURCE CFD-DEM FOR RESOLVED PARTICLE-FLUID INTERACTION

Alice HAGER[1*], **Christoph KLOSS**[1, 2], **Stefan PIRKER**[1, 3] **and Christoph GONIVA**[1, 2]

[1] Christian Doppler Laboratory on Particulate Flow Modelling
[2] DCS Computing GmbH
[3] Institute of Fluid Mechanics and Heat Transfer
Johannes Kepler University, Altenbergerstraße 69, 4040 Linz
AUSTRIA
*Corresponding author, E-mail address: alice.hager@jku.at

ABSTRACT

In the following paper we present a parallelized resolved method for the simulation of the dynamics of immersed bodies within fluids. The algorithm uses the so called Fictitious Domain Method (FDM) and combines the Lagrangian Discrete Element Method (DEM) for the tracking of the bodies and a Computational Fluid Dynamics (CFD) method for the calculation of the flow and pressure field of the fluid phase. In a first step the CFD-calculation is carried out, disregarding the solid bodies. Afterwards, the velocity information from the bodies is included and the force, the fluid imposes onto the bodies, is calculated. The last step consists of a correction-operation which ensures the fulfilment of the conservation equation. Dynamic local mesh refinement is used in the area of the bodies in order to keep the number of fluid cells to a minimum.

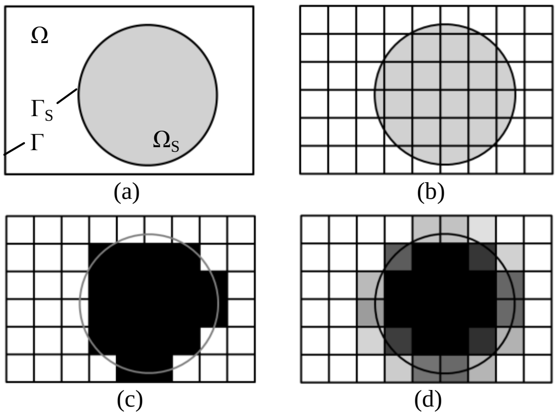
The CFD-DEM coupling is realized within the Open Source framework CFDEMcoupling (www.cfdem.com), where the DEM software LIGGGHTS (www.liggghts.com) is linked against an OpenFOAM®-based CFD solver. While both LIGGGHTS and the CFD solver could already be used for parallel computations, only a recent improvement of the algorithm permits the fully parallel computation of resolved problems. This parallelization permits the treatment of large-scale problems. The enclosed validation and application examples show the dynamics of the flow around settling and rotating spheres as well as an investigation of the settling of spheres regarding the Boycott effect.

NOMENCLATURE

p	pressure
$\mathbf{u}$, $\mathbf{u}_p$	velocity, particle velocity
$\boldsymbol{\omega}^*$, $\boldsymbol{\omega}$	(dimensionless) angular velocity of the sphere
$\mathbf{U}$	translational velocity of the sphere
$\mathbf{r}$	position vector
ρ, ρ_F	density, fluid density
μ	dynamic viscosity
Ω, Ω_S	computational domain, solid domain
Γ, Γ_s	domain/body boundary
F_D, F_L	drag and lift force
A	cross section area of the sphere

INTRODUCTION

In the field of particle-fluid interactions a variety of methods combining Eulerian and Lagrangian approaches find application. Especially for small particles unresolved methods involving a void fraction field for capturing the impact of the particulate phase onto the fluid phase and vice versa are applied (cf. Zhou et al., 2010). In contrast to that, resolved methods are used for the representation of particles or bodies with a, compared to the remaining geometry, large diameter. In the present case the dynamics of the fluid around the bodies are calculated with a Computational Fluid Dynamics (CFD) method while the motion of the moving objects is determined with a Discrete Element Method (DEM) approach. Fig. 1 shows an example of a problem that can be tackled with the present method: the original problem consists of a body within a fluid domain (Fig. 1 (a)). In order to treat the problem numerically, the fluid domain is meshed disregarding the existence of the solid areas (Fig. 1 (b)). Finally the object(s) have to be located within the mesh, as can be seen in Fig. 1 (c) and (d). Details on the last matter can be found in the next section.



(a) (b)
(c) (d)

Figure 1: Body within the computational domain (a), meshing of the whole domain independently of the position of the body (b), stair step representation (c) and smooth representation (d).

For the sake of clarity, all theoretical explanations were and will be explained for a single-body problem. Yet all conclusions and assumptions can be extended to problems involving an arbitrary number of objects. Besides a number of validation and application examples, this paper comprises a description of the applied method and a collection of the latest algorithmic improvements. A serial version of this method has been presented and validated before (cf. e.g., Hager et al., 2011a). Further details on the coupling-software CFDEMcoupling in general can be found in various publications by the authors (e.g., Goniva et al., 2010 or Kloss et al., 2010).

MODEL DESCRIPTION

The computational method

The particularity of this combined method lies in the separate treatment of fluid and solid phase: in a first step the dynamics of the fluid phase is calculated, neglecting the existence of the bodies, using data from the previous time step as initialization. The equations describing the flow field are the incompressible Navier Stokes equations, consisting of the momentum equations (1) and the continuity equation (2), together with boundary (3) and initial conditions (4):

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \mu\nabla^2\mathbf{u} \quad \text{in } \Omega \qquad (1)$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in}\,\Omega \qquad (2)$$

$$\mathbf{u} = \mathbf{u}_\Gamma \quad \text{on}\,\Gamma \qquad (3)$$

$$\mathbf{u}(x, t=0) = \mathbf{u}_0(x) \quad \text{in}\,\Omega \qquad (4)$$

With this new velocity and pressure information the force acting onto the objects can be evaluated. Following Shirgaonkar et al., 2009, the force exerted on the particles consists of a pressure and a viscous term:

$$f = -\nabla p + \nu\nabla^2\mathbf{u}. \qquad (5)$$

This force term (5) stems from the transformation of the second of two interface conditions:

$$\mathbf{u} = \mathbf{u}_i \text{ on } \Omega_S \quad \text{and}\, \sigma\cdot\hat{\mathbf{n}} = \mathbf{t}_{\Gamma_s} \text{on}\, \Gamma_s\,. \qquad (6)$$

It describes the stress at the interface between the solid body and the fluid. This term together with contributions for buoyancy, gravity, particle-particle collisions and particle-wall interactions is finally used to determine the motion of the bodies with LIGGGHTS. The whole palette of models implemented in the DEM-code can be used to depict the behaviour of the bodies.

In the next step, the particle-velocity is incorporated into the velocity field of the fluid phase. The correction of the velocity field takes place in those cells, where the particle is located, as demanded by the first equation of (6). The correction operation causes a violation of the mass conservation, so the concluding step of the procedure consists of a correction operation. These steps are repeated for each time step. CFD and DEM calculation in many cases require different time step widths, which can easily be taken into account, as the number of time steps between the coupling is user-defined.

Similar approaches for determining the motion of immersed bodies can for example be found in Glowinski et al., 1998, Patankar et al., 2000 or Uhlmann, 2005; Haeri and Shrimpton, 2012, give an overview over several existing methods together with a number of application examples.

Representation of the bodies within the fluid phase

Fig. 1 (c) and (d) show two different ways of representing the bodies within the fluid mesh: Fig. 1 (c) shows a case, where each individual cell is considered either fluid or solid. This sharp distinction is very fast, but causes a slight numerical mal-representation of the bodies. This effect can be diminished by a rather small mesh size in the area of the solids. The method of dynamic local mesh refinement helps to avoid unnecessarily fine meshes over the whole domain by only resolving those areas, where a body is located. For more details on this topic see, e.g., Hager et al., 2011b. Nevertheless, in some cases refinement cannot eliminate the occurring troubles: due to the sharp interfaces certain areas of the mesh are isolated from the rest of the domain, even though in the original problem the region is connected with the remaining fluid area. This leads to numerical instabilities. Fig. 2 shows a problem where this locking appears, some of the “cut-off” areas are marked by dark boxes.

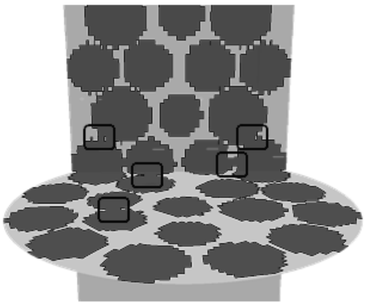


Figure 2: Closing of gaps due to the mal representation of particles.

As a remedy, the present solver comprises a new particle location model, which takes the degree to which the very cell is covered by solid into account. The chosen method is discussed in the following section. In Fig. 1 (d) the cells are coloured according to the amount of solid volume they occupy. Once this information is available, it can be used for correcting the velocity with respect to both fluid and solid accordingly. Furthermore, this information also plays a role for the evaluation of specific force terms acting onto the body.

Computation of the void fraction inside a cell

When determining the volume of the part of the cell, covered by solid, a balance between computational costs and accuracy has to be found. Using classical stochastic methods for each cell is too expensive, especially for a growing number of particles. A possible representation which assumes cubic mesh cells is the following one: The algorithm first decides whether the cell is close enough to the surface in order to be only partly filled. If this case occurs, all 8 vertices are checked. If the vertex lies inside the body, a contribution of 1/8 is added to the volume fraction of the cell. Otherwise the intersection point of sphere surface and the line connecting vertex and cell centre is computed. The relative length of the line from centre of intersection point times 1/8 is the added contribution for this case. Fig. 3 shows the method for a 2D cell.

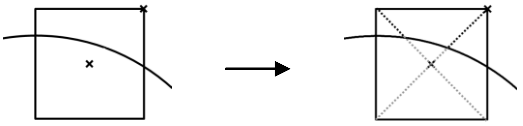


Figure 3: Determination of the volume fraction

The restriction to cubic mesh cells can easily be lifted for general cases, but as for the present applications this condition is met automatically, this case will not be discussed here.

Determination of the void fraction – Validation

In order to show the accuracy of the void fraction model, the algorithm was implemented in Matlab and compared to the results gained by a Monte Carlo method. For getting a number of comparative cases, the cell centre is moved on a line that draws a certain angle α with the horizontal (cf. Fig. 4)

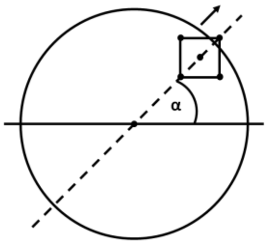


Figure 4: Comparative cases for the validation of the void fraction model

Fig. 5 shows the relative errors made by the smooth void fraction model. Both results good accordance, especially as the 3D case comprises 8 control points only.

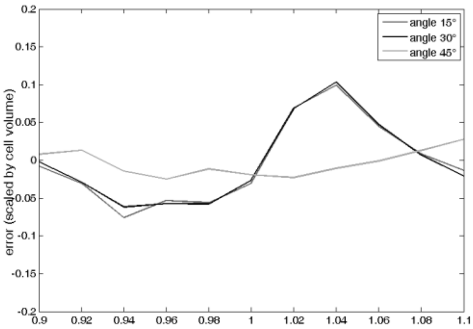


Figure 5: Deviation of the volume from Monte Carlo results for α = 15°, 30° and 45°.

IMPLEMENTATION

Parallelization of the method

Investigations showed that at least 8 cells per particle diameter are necessary for gaining accurate results. For problems with a bigger number of particles this results in large meshes and a parallelization of the code is required for obtaining results within reasonable time.

Both OpenFOAM® and LIGGGHTS can be run in parallel using MPI. Yet the computation does not only involve a CFD-solver and a LIGGGHTS calculation but also a number of sub-models accounting for several coupling-issues such as force calculation and location of the bodies within the fluid phase. Especially the parallelization of the latter one requires some adjustments in order to depict the particles correctly when passing from one processor to the next.

In the basic version, the body-location algorithm starts from the cell, which contains the centre of the object and checks the neighbouring cells recursively. Objects about to traverse a processor boundary and thus located on two processors at once are, therefore, represented incorrectly. In a new approach, which is sketched in Fig. 6, the location algorithm uses additional satellite points for starting the location routine. These points are situated on the surface of the sphere and can be distributed arbitrarily densely. The correct depiction of the particle at each time step is especially important for the force calculation: even the misrepresentation for a couple of time steps has a huge effect on the body motion.

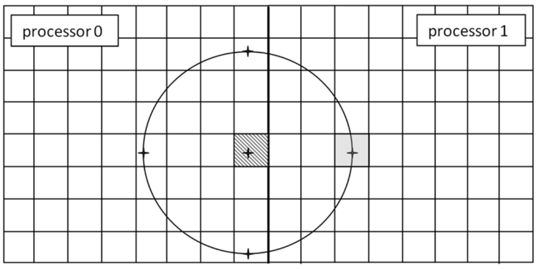


Figure 6: Particle location process in the parallelized model – grey shaded cells mark the starting positions of the locate algorithm on the two processors respectively.

Parallelization – Benchmark

In order to determine the efficiency of the parallelization, test runs were launched for different numbers of processors. Tab. 1 gives an overview for the scaling behaviour of a problem with approximately 130,000 cells. The computational time is scaled with the time elapsing for the calculation on two processors.

number of processors	computational time
2	1.00
4	0.60
8	0.22
12	0.10

Table 1: Performance of the behaviour of a problem with approx. 130.000 cells

The resolved method is designed for problems comprising a limited number of bodies (O(10^3)). The reason for this limitation lies in the fact that each particle requires a certain resolution, which leads to huge mesh sizes. So clearly the limiting factor regarding case-size lies on the CFD side. Classical CFD-DEM problems can easily handle problems with O(10^6) particles and more.

VALIDATION AND APPLICATION

The presented method can be used for a wide field of problems. While with the first examples the coupled method is validated, the last example is a typical application of a coupled CFD-DEM solver.

Flow around a fixed sphere

In the following case we consider a fixed sphere in a fluid flow at Reynolds numbers between 100 and 1000. For Re = 100 the fluid passes the object without any instabilities and shows symmetric behaviour in the wake downstream of the body. For increasing flow velocities, i.e. for higher Reynolds numbers the regime changes, until at a Reynolds number of 300 a vortex street develops. This observation is in accordance with results by Sakamoto and Haniu, 1990. The drag force acting onto the sphere can be used for determining the dimensionless drag coefficient by using the relation

$$C_D = \frac{F_D}{\frac{1}{2}\rho_F \mathbf{u}^2 A} \quad . \tag{8}$$

Fig. 7 shows the drag coefficient on the sphere for this case, Fig. 8 the corresponding velocity field. The structures in the latter figure depict the spanwise velocities, i.e. the instabilities in the flow.

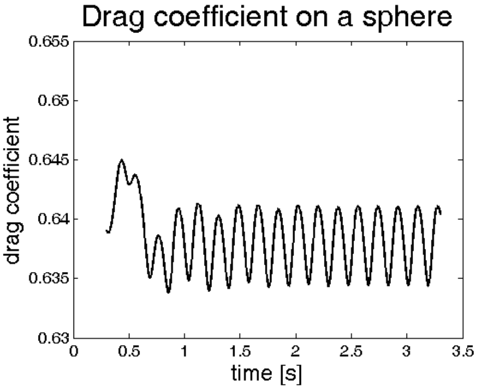


Figure 7: Drag coefficient on a sphere for Re = 300

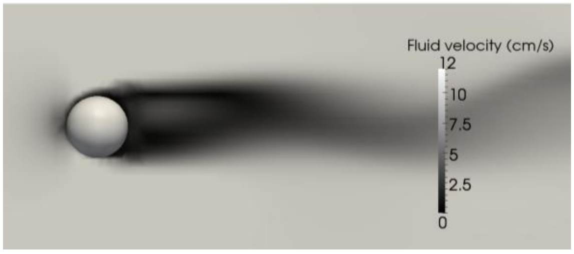


Figure 8: Velocity around a sphere at Re = 300

In Fig. 9 drag coefficients gained from the simulations are compared against values from literature (cf. Böswirth, 2007). The area of interest is marked by a box; the bold black line marks the simulation results.

Flow around a rotating sphere

Generally the motion of a rigid sphere can be described by

$$\mathbf{u}_p = \mathbf{U} + \boldsymbol{\omega} \times \mathbf{r}, \tag{7}$$

meaning it consists of a translational and a rotational component. While previous publications focused on the validation of effects in context with the translational component which dominates in many cases, we here concentrate our attention on the rotation.

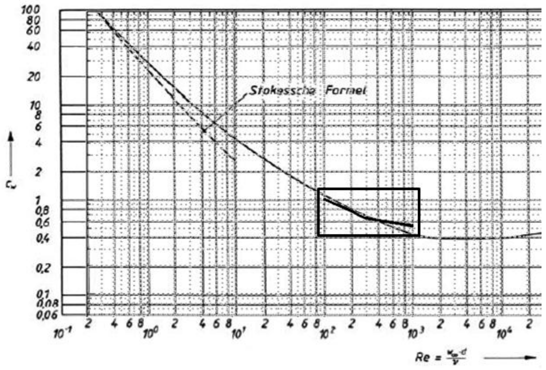


Figure 9: Drag coefficient on a sphere for different Reynolds numbers, comparison simulation results (---) and literature (cf. Böswith, 2007, p. 217).

Due to the rotation of the sphere the fluid is accelerated at one side of the sphere, which, according to Bernoulli, causes a pressure drop. The different pressures on the two opponent sides of the body result in a lift force, termed a Magnus force. Fig. 10 is a screenshot from one of the calculations; it shows the streamlines around the rotating sphere. In the area of the particle the velocity vectors depict the rotation.

For Reynolds numbers between 10 and 140 Oesterlé and Bui Dinh, 1998, derived the following correlation between Reynolds number and lift coefficient:

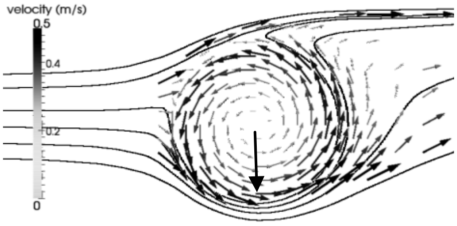


Figure 10: Flow around a rotating sphere, Magnus force acts on sphere (vertical arrow)

$$C_L \approx 0.45 + (2\omega^* - 0.45)\, e^{-0.075\omega^{*0.4}\,\mathrm{Re}^{0.7}}, \tag{9}$$

where the lift coefficient C_L is calculated similar to C_D, using the lift force F_L instead of F_D:

$$C_L = \frac{F_L}{\frac{1}{2}\rho_F \mathbf{u}^2 A} \quad . \tag{10}$$

The dimensionless angular velocity $\boldsymbol{\omega}^*$ is given by r$\boldsymbol{\omega}$/$\mathbf{u}$, where r denotes the radius of the sphere this time. In Fig. 11 results gained from simulations are compared to the relation given by Oesterlé and Bui Dingh. While for lower dimensionless angular velocities deviations can be seen, there is a good accordance in the higher region.

Settling of two spheres

In further investigations, the settling behaviour of two spheres was analysed with the present method. The cases were chosen according to settings in Glowinski, 2001 and Uhlmann, 2005 for being able to directly validate the results.

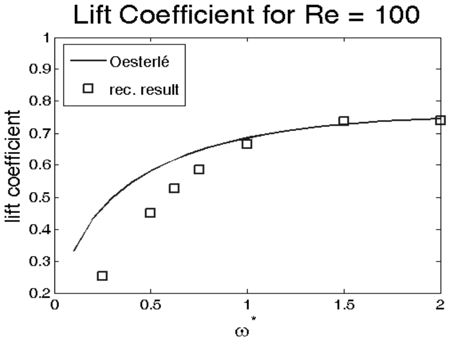


Figure 11: Comparison of the lift coefficients, □ recent results from the simulations, --- predicted value (Oesterlé and Bui Ding, 1998)

The two spheres of diameter 0.167 cm are placed in a cuboid box of height 4 cm and a square base of length 1 cm. Their centres are initialized central with respect to the base and at the heights 3.16 and 3.5 respectively. The gravity vector points towards the negative z-axis, i.e. is given by (0 0 -981). The simulations were carried out for different sphere densities (1.14 and 1.5 g/cm³). The viscosity of the fluid is given with 0.01 cm²/s. Fig. 12 shows the comparison of the settling velocities from the simulation compared to the result by Glowinski.

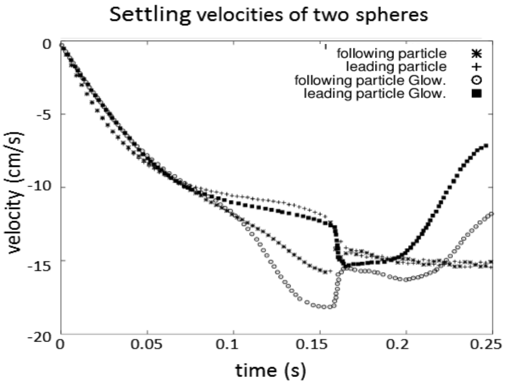


Figure 12: Settling of two spheres with density 1.5 g/cm³, present simulations vs. results by Glowinski.

During the settling process of the spheres three different states can be observed: first the objects start their descent "in parallel", meaning that they do not influence each other. As soon as the upper particle reaches the wake of the lower one, it gets accelerated, and while approaching the leading particles, it influences its motion as well. After colliding, the behaviour of the two spheres becomes chaotic until they finally reach the bottom of the domain. These states are termed "drafting, kissing and tumbling." For a more detailed study of this case see Hager et al., 2011a, b.

The Boycott effect

The investigation of the settling behaviour of particles within a suspension showed that particles settling in a tilted tube sediment much faster than those in a vertical tube, cf. Boycott, 1920, Ungarish, 1993. This effect is evoked by the fact that the particles approach the sloping wall and leave an area with pure liquid or only few particles above them. The region with the higher solid-concentration exhibits a higher density than the remaining fluid which leads to a circulating flow in the tube and thus a faster settling of the solids. Fig. 13 gives a schematic representation of the settling process after a certain time: the upper region of the suspension is already cleared of particles, while in the lower region one can distinguish between a dilute and a dense suspension. The (purely imaginary) border between the two regions is marked by a line.

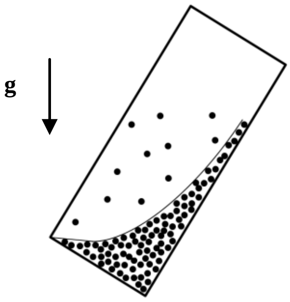


Figure 13: Settling of particles within a tilted tube, dense suspension, dilute suspension and pure fluid.

An extensive investigation of the effect can be found in Xu and Michaelides, 2005.

The Boycott effect – Settling of spheres

In order to reproduce this effect, the following testcase was defined: A cuboid with quadratic base of length 1 cm and with a height of 5 cm was chosen. A dense packing of 50 particles with a diameter of 0.167 cm was generated above the height of 4 cm and released at t = 0. For one case the gravity vector points straight in the direction of the negative z-axis, (i.e. g = (0 0 -981) cm/s^2, for the comparable case the gravity vector is shifted by an angle of 45° (i.e., g = 981*(-0.707 0 -0.707)). Fig. 14 shows a screenshot from the velocity fields of the two calculations at the same time step. For the straight case the main velocities are those in the area of the bodies, while for the tilted case the development of the circulation of the fluid becomes visible. This flow accelerates the settling of the spheres.

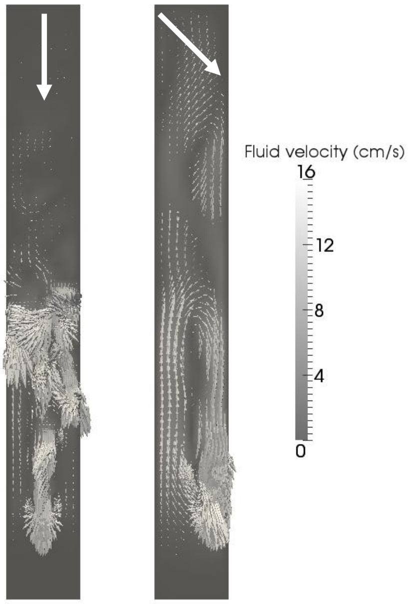


Figure 14: Velocity field with glyph vectors – body motion dominates straight case (left), circulation develops for tilted case (right).

Fig. 15 (a) shows the initial configuration of the discussed cases. Furthermore Fig. 15 comprises screenshots of the states at two different times for the case with the vertical and the shifted gravity vector respectively. For t = 1.1 (Fig. 15(d) and (e)) the discussed effect shows best: while the sedimentation process is still ongoing for the straight case, particles in the tilted case are already settled.

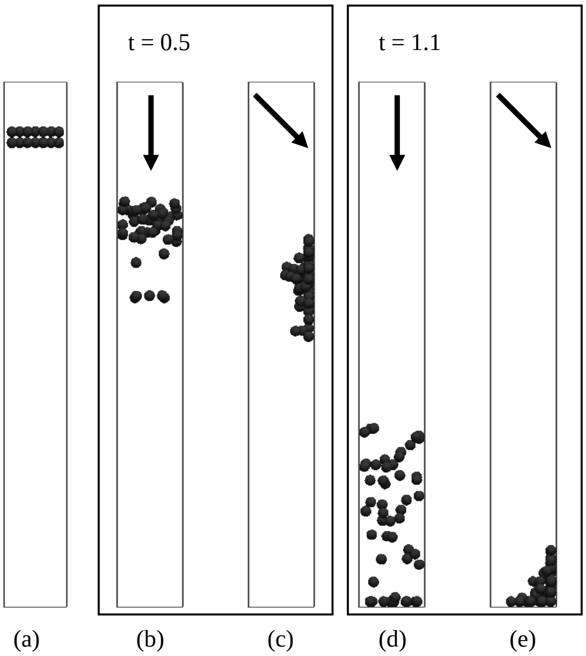


Figure 15: Initial position of the spheres (a), settling of the packing at two different time steps. The arrows point in the direction of the gravity vector.

CONCLUSION

The present paper contains a description of the resolved CFD-DEM method used, focusing on the representation of the bodies within the fluid. In the implementation-section the authors elaborate on the concept of the parallelization. The range of applications reaches from a detailed, resolved depiction of the impact, fluid has on single settling, rotating or fixed particles to multi body problems. With the aid of the parallelization, which allows the treatment of some hundreds or thousands of objects, it was possible to demonstrate the Boycott effect. For a significantly higher number of particles the resolved approach did not prove successful. In a next step merging this resolved method with an unresolved method for tackling massively polydisperse problems would be of interest.

REFERENCES

BÖSWIRTH, L. (2007), “Technische Strömungslehre”, Ausgabe 7, Vieweg Verlag, ISBN: 978-3-8348-0272-9

BOYCOTT, A. E., (1920), “Sedimentation of blood corpuscles”. *Nature* **104**, 532.

CFDEMcoupling, (2012), “CFDEM – Open Source CFD, DEM and CFD-DEM”, URL: www.cfdem.com

GLOWINSKI, R., PAN, T.-W., HESLA, T.I., JOSEPH, D.D. and PERIAUX, J., (1998), “A fictitious domain method with distributed Lagrange multipliers for the numerical simulation of particulate flow”, *Contemporary Mathematics*, **218**, 121-137

GLOWINSKI, R., PAN, T.-W., HESLA, T.I., JOSEPH, D.D. and PERIAUX, J, (2001). “A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow”, *J. Comput. Phys*, **169**, 363-426

GONIVA, C., KLOSS, C., HAGER, A. and PIRKER, S., (2010), “An Open Source CFD-DEM perspective”, *Proc. 5th OpenFOAM Workshop*, Chalmers, Gothenburg, Sweden, June 21-24.

HAERI, S. and SHIRMPTON, J.S., (2012), “On the application of immersed boundary, fictitious domain and body conformal mesh methods to many particle multiphase flows”, *Int. J. Multiphase Flow*, **40**, 38-55

HAGER, A., KLOSS, C. and GONIVA, C., (2011a), “Towards an efficient immersed boundary method within an Open Source Framework”, *Proc.12th International Conference on Multiphase Flow in Industrial Plants*, Ischia (Napoli), Italy, September 21-23

HAGER, A., KLOSS, C. and GONIVA, C., (2011b), “Efficient realization of a resolved CFD-DEM method within an Open Source framework”, *Proc. Open Source International Conference 2011*, Paris-Chantilly, France, November 3-4.

KIM, D. and CHOI, H., (2002), “Laminar Flow past a sphere rotating in the streamwise direction”, *J. Fluid Mech.*, **461**, 365-286

KLOSS, C., GONIVA, C. and PIRKER, S., (2010), “Open Source DEM and CFD-DEM with LIGGGHTS and OpenFOAM”, *Proc. Open Source CFD International Conference*, Munich, Germany, November 4-5.

LIGGGHTS, (2011), “LAMMPS improved for general granular and granular heat transfer simulations”, URL: www.liggghts.com

OESTERLÉ, B and BUI DINH, D., (1998), “Experiments on the lift of a spinning sphere in a range of intermediate Reynolds numbers”, *Experiments in Fluids*, **25**, 16-22

OpenCFD Ltd., (2009), “OpenFOAM – The Open Source CFD-toolbox”, URL: 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*, **26**, 1509-1524

SAKAMOTO, H. and HANUI, H. (1990), “A study on vortex shedding from spheres in a uniform flow”, *J. Fluids. Eng.*, **112**, 386-392

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

UHLMANN, M., (2005). “An immersed boundary method with direct forcing for the simulation of particulate flows.”, J. Comput. Phys., 209, 444-476

UNGARISH, M., (1993), “Hydrodynamics of suspensions”, Springer Verlag, ISBN 3-540-54762-2

XU, Z.-J. and MICHAELIDES, E.E., (2005). “A numerical simulation of the Boycott effect”, *Chem. Eng. Comm.*, **192**, 532-549

ZHOU, Z.Y., KUANG, S.B., CHU, K.W. and Yu, A.b., (2010), “Discrete particle simulation of particle-fluid flow: model formulations and their applicability”, *J. Fluid Mech*., **661**, 482-510