OCTREE-GENERATED VIRTUAL MESH FOR IMPROVED CONTACT RESOLUTION IN CFD-DEM COUPLING

Ondřej Studeník 1, Martin Kotouč Šourek 1,2, Martin Isoz 1,3
1 Department of Mathematics, Faculty of Chemical Engineering, University of Chemistry and Technology, Technická 5, Prague 166 28, Czech Republic
2 Department of Chemical Engineering, University of Chemistry and Technology, Technická 5, Prague 166 28, Czech Republic
3 Czech Academy of Sciences, Institute of Thermomechanics, Dolejškova 5, Prague 182 00, Czech Republic

Abstract

The present work is focused on improving the efficiency of a computational fluid dynamics (CFD) – discrete element method (DEM) solver allowing for computations with non-spherical solids. In general, the combination of CFD and DEM allows for simulations of freely moving solid particles within a computational domain containing fluid. The standard approach of CFD-DEM solvers is to approximate solid bodies by spheres, the geometry of which can be fully defined via its radius and center position. Consequently, the standard DEM contact models are based on an overlap depth between particles, which can be easily evaluated for a sphere-sphere contact. However, for a contact between two non-spherical particles, the overlap depth cannot be used and has to be replaced by the more general overlap volume. The precision of the overlap volume computation is (i) crucial for the correct evaluation of contact forces, and (ii) directly dependent on the computational mesh resolution. Still, the contact volume evaluation in DEM for arbitrarily shaped bodies is usually by at least one order of magnitude more demanding on the mesh resolution than the CFD. In order to improve the computational efficiency of our CFD-DEM solver, we introduce the concept of an OCTREE-based virtual mesh, in which the DEM spatial discretization is adaptively refined while the CFD mesh remains unchanged.

Keywords: HFDIB-DEM, CFD, DEM, washcoating, OpenFOAM.

1 Introduction

Processes with a solid phase dispersed in a fluid are omnipresent in both nature and industry. A few examples from the chemical engineering practice include the deposition of an active catalytic layer into a monolith structure [1], sedimentation [2] or fluidization [3]. In these processes, the solid particles are distributed so densely, that they come into contact with each other and with the walls of the device. Furthermore, the fluid and solid phases are mutually affected. Nowadays, such processes are usually described using empirical models and validated correlations [2, 3]. However, these methods may not always be sufficient.

For example, if a detailed and localized description of both solid and fluid phases is required, the processes may be simulated via a combination of the computational fluid dynamics (CFD) and discrete element (DEM) methods. Connecting CFD and DEM allows to effectively model (i) a fluid flow, (ii) a particle motion, and (iii) solid-solid contacts, taking into account the mutual influence between the fluid and the solid phases. With respect to DEM, the vast majority of the current CFD-DEM solvers approximates arbitrarily shaped particles, which are prevalent in nature, either directly as spheres or as agglomerates of spheres [4]. Alternatively, some codes allow for other primitive shapes [4]. Still, all particle shapes commonly used in DEM can be described via a small set of parameters.

Utilizing geometries describable through a limited number of parameters greatly facilitates the evaluation of contact forces. For example, in the case of the soft DEM [5], which allows for small particle-particle overlaps, the contact forces are determined based on the overlap depth \( \delta \). For the contact between two primitive bodies, the overlap depth can be readily computed, e.g. a sphere-sphere contact in which \( \delta = r_1 + r_2 - d_{12} \), where \( r_1 \) and \( r_2 \) are the spheres radii and \( d_{12} \) the distance between their centers.
However, when evaluating the influence of the particle geometry on the system behavior, the approximation by a primitive body may not possible and arbitrarily-shaped bodies need to be taken into account. The geometry of such bodies cannot be easily parametrized and has to be defined by the complete body surface. Consequently, the contact forces can no longer be based on $\delta$ as its definition for the contact between non-convex bodies becomes ambiguous. Still, the solid surface can be provided in a discrete form, for example in a stereolithography (STL) file format, and the overlap depth can be generalized to the overlap volume.

In the present work, we concentrate on improving the efficiency of a CFD-DEM solver usable for simulations with arbitrarily shaped solid bodies. The inclusion of the moving solid phase in the CFD computational domain is performed using the immersed boundary method variant referred to as HFDIB (Hybrid Fictitious Domain-Immersed Boundary) [6, 7]. Utilizing HFDIB to project solids onto the CFD mesh (i) eliminates the need to adapt the mesh for the movements of solids, but (ii) causes a discrepancy between the exact solid body shape and its projection onto the CFD mesh.

The quality of the solid bodies projection onto the CFD mesh may be improved by a dynamic refinement of the CFD mesh in the vicinity of the solid phase. However, with such an approach, we lose the independence of the CFD grid on the bodies motion, which slows down the computation. An alternative proposed in this work is to create a local and purely DEM mesh only if a solid-solid contact is possible or occurring. This DEM mesh is virtual with respect to the global CFD spatial discretization, which allows for increase in the DEM accuracy without affecting the CFD calculation complexity. Moreover, the cost of the virtual DEM mesh construction is further reduced by employing the Octree algorithm [8].

The contribution is structured as follows. First, we give the fundamentals of our CFD-DEM solver and point out the specifics of computations with arbitrarily-shaped solids. Next, the virtual mesh implementation is described in detail. Afterwards, the new virtual mesh concept is verified utilizing a contact of two spheres and compared to the results obtained (i) from a sphere-based DEM, and (ii) utilizing a CFD mesh dynamic refinement. Finally, the new DEM implementation capabilities are illustrated on the contact of two generally-shaped bodies solved using the virtual mesh and dynamic CFD mesh refinement.

## 2 Used computational methods

CFD-DEM-based solvers for particle-laden flows need to resolve two main challenges. First, to allow for efficient computations, the solids need to be included in the computational domain without the need for frequent domain remeshing. The second problem poses the solver architecture itself, i.e. coupling of CFD and DEM solver parts. In the present work, we opted for a monolithic solver architecture meaning that both the CFD and DEM are implemented within the same code, the OpenFOAM finite volume (FV) C++ library [9], which necessitated a custom DEM solver development.

### 2.1 Inclusion of particles into the computational domain

Let $\Omega \subset \mathbb{R}^3$ be the spatial computational domain and $\Omega^h$ its finite volume discretization. Furthermore, let $\Omega_\alpha$ and $\Omega_\ell$ be the parts of $\Omega$ occupied by the solid and fluid phases, respectively. The commonly adopted approach in CFD is to exert the influence of $\Omega_\alpha$ on the flowing fluid by subtracting it from $\Omega$ and putting $\Omega := \Omega_\ell$. However, for the case of $\Omega_\alpha$ freely moving within $\Omega$, such an approach would require frequent reconstruction of $\Omega^h$, i.e. remeshing.

In order to circumvent the need for frequent remeshing, the computational domain is kept as $\Omega = \Omega_\alpha \cup \Omega_\ell$ and the $\Omega$ division into $\Omega_\alpha$ and $\Omega_\ell$ is provided via a hybrid fictitious domain-immersed boundary method (HFDIB). For example, an incompressible flow of a Newtonian fluid is assumed to be governed by equations

$$
\mathcal{M}(\mathbf{u}) = -\nabla \tilde{p} + \mathbf{g} + \mathbf{s} \quad \text{and} \quad \nabla \cdot \mathbf{u} = 0, \quad \mathcal{M}(\mathbf{u}) = -\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\nu \nabla \mathbf{u}) \quad (1)
$$

where $\mathbf{u}$ is the fluid velocity, $\nu$ kinematic viscosity, $\tilde{p}$ kinematic pressure, $\mathbf{g}$ the gravitational acceleration, and $\mathbf{s}$ is a yet unspecified force required to prevent the flow from entering $\Omega_\alpha$ and to enforce the required boundary conditions on $\Gamma_{sf}$, the boundary between $\Omega_\alpha$ and $\Omega_\ell$. 
For the particular case of HFDIB, the forcing term is defined as follows,

\[ s = \text{ceil}(\lambda) \tilde{s}, \quad \tilde{s} = \mathcal{M}(u_{ib}) + \nabla \tilde{p} - g, \quad \lambda = \begin{cases} 0 & \text{in } \Omega_f \\ 1 & \text{in } \Omega_s \\ \tilde{\lambda} \in (0, 1) & \text{in } \Gamma_{sf} \end{cases} \tag{2} \]

where \( \lambda \) is an indicator function defining the position of the immersed bodies in the computational domain \( \Omega \), \( u_{ib} \) is the velocity of the moving solid phase.

Note that the exact definition of \( s \) depends on the approach used to enforce the boundary conditions on \( \Gamma_{sf} \). Because \( \Gamma_{sf} \) cannot be assumed collocated with \( \Omega^h \), the boundary conditions enforcing the need to be based on interpolations. For details on the specifics of HFDIB, the reader is referred to [6, 7].

### 2.2 Movement of the bodies and contact resolution

With the solid phase immersed via HFDIB, the next step is the solution of its motion. The movement of individual solids \( B_i, i = 1, \ldots, n \) is described via the discrete element method (DEM). DEM is a finite difference method based on Lagrangian solution of Newton’s second law of motion,

\[
m_i \frac{d^2 x_i}{dt^2} = \sum_{j=1}^{N_f} f_i^j, \quad I_i \frac{d\omega_i}{dt} = \sum_{j=1}^{N_i} t_i^j, \tag{3}\]

where \( m_i \) stands for the mass of \( B_i \) and \( x_i(t) \) for its center of mass position at time \( t \), \( \omega_i \) is the body angular velocity and \( I_i \) is the matrix of its inertial moments. The sums on the right-hand sides of equations (3) represent all forces \( f \) and torques \( t \) acting on the body \( B_i \), respectively.

![Figure 1: Fundamentals of notation in CFD-DEM contact solution for arbitrarily shaped solids](image)

(a) projection of bodies \( B_i \) and \( B_j \) onto CFD mesh \( \Omega^h \), surface cells \( \partial B_i \) and \( \partial B_j \) of \( B_i \) and \( B_j \) are shown in green and red, respectively, (b) overlap volume \( V^o_{ij} \) between the two solids in contact, composed of the contact cells, (c) edge cells \( E_{ij} \) and contact normal \( n \).

The standard forces and torques taken in the account are (i) gravity and buoyancy, (ii) coupling between the solid and fluid motion via the forcing term \( s \) computed as given in (2), and (iii) contact. For specifics of the implementation of the first two, the reader is referred to [6]. In the present work, we will focus solely on the contact, namely on the normal contact force \( f_{c,n} \), which is fundamental for any contact solution. The \( f_{c,n} \) computation is handled by soft particle model that allows for particles overlap and the force is evaluated based on the overlap magnitude.

The commonly used approach is to simplify evaluation of the overlap by replacing the general shape of a particle with a simple shape, see e.g. the Hertz model \([5]\) in which particles are approximated as spheres. For spheres, the overlap can be evaluated according to the overlap depth \( \delta^o \) as indicated in the Introduction. However, such a simplification is not possible when the behaviour of a generally shaped body is of interest. In that case, \( \delta^o \) has to be replaced by a more general concept of the overlap volume \( V^o_{ij} \) of bodies \( B_i \) and \( B_j \). Then, the force \( f_{c,n} \) is evaluated as

\[
f_{c,n} = \frac{Y^* V^o_{ij}}{l_c} n, \quad \frac{1}{Y^*} = \frac{1}{Y_i} + \frac{1}{Y_j} , \quad V^o_{ij} = \int_{B_i \cap B_j} \lambda dV , \tag{4}\]

where \( Y^* \) is the effective modulus for the contact problem, \( Y_i \) and \( Y_j \) are the moduli of the bodies \( B_i \) and \( B_j \).
Figure 2: Fundamentals of the virtual mesh concept, (a) solids \( B_i \) and \( B_j \) projected onto \( \Omega^h \), (b) constructed virtual mesh (in cyan), (c) detail of the virtual mesh close to an edge cell (in yellow), (d) encoding of the situation from (c) in binary lists isInBody\(_i\) (green), isInBody\(_j\) (red) and isInBoth (white on black), (e) identified edge cells and contact normal.

where \( Y \) stands for Young’s modulus, \( \mathbf{n} \) for the contact normal, \( l_c \) represents characteristic length of contact, and \( B_i \cap B_j \) are the cells shared between the bodies \( B_i \) and \( B_j \). For detailed description of the model (4) and its physical motivation, see [10].

For a general case, the estimation of \( V_{ij}^o \) is not straightforward. Nonetheless, in our case, the CFD domain discretization \( \Omega^h \) may be utilized as a basis for \( V_{ij}^o \) computation. In particular, \( V_{ij}^o \) may be computed via numerical approximation if the integral (4) is not evaluated precisely, which impairs the \( f_{ij}, \mathbf{n} \) computation. Furthermore, when \( B_i \) and \( B_j \) share only a single cell, the normal \( \mathbf{n} \) cannot be determined, which may lead to nonphysical contact behavior.

2.3 Virtual mesh fundamentals

As indicated above, the accuracy of \( f^{c,n} \) computation strongly depends on \( \Omega^h \) resolution, which influences the quality of \( V_{ij}^o \) and \( \mathbf{n} \) estimates. To increase the quality of arbitrary shaped solids representation in DEM computations without the need to modify \( \Omega^h \), we propose an Octree-leveraging algorithm, called virtual mesh, which enables us to improve the accuracy of \( V_{ij}^o \) and \( \mathbf{n} \) estimation.

The algorithm itself is given in Alg. 1. Furthermore, its fundamental concepts are illustrated in Fig. 2. For contact between the bodies \( B_i \) and \( B_j \), the virtual mesh algorithm is based on uniformly dividing each contact cell, i.e. a cell in \( B_i \cap B_j \), in \( n^3 \) sub-volumes \( \Xi = \{ \xi_r \}_{r=1}^n \) and effectively comprises two parts, (i) \( V_{ij}^o \) computation, and (ii) detection of edge sub-volumes \( \mathcal{E}_{ij}^\xi \) that are subsequently used to improve the \( \mathbf{n} \) estimate.

Overlap volume estimation  Note that in our solver, the solids are defined by STL representations of their surface. Because of the loss of information caused by projecting the STL-defined solids onto \( \Omega^h \) via the \( \lambda \) field, it may happen that a cell \( \Omega^h_{ij} \) is in both \( \partial B_i \) and \( \partial B_j \); yet, \( B_i \) and \( B_j \) are not in contact. Nevertheless, even such a cell needs to be included in the contact cells \( \mathcal{E}_{ij}^\xi \) to be treated by the virtual mesh algorithm. On the other hand, there is no need to use virtual mesh for cells inside both bodies. The created sub-volumes are depicted in Fig. 2a,b. After the decomposition of \( \Omega^h \) into \( \Xi \) sub-volumes, we identify all the \( \xi \in \Xi \) common for both bodies and
Algorithm 1 Algorithm describing creation of virtual mesh and its application for overlap volume computation and for edge sub-volumes detection

Require: list of contact cells, \( L_{ij} \), level of cell decomposition, \( n \)

1: \( V_{ij}^o = 0, \xi_{ij} = [ ] \); \( \triangleright \) initialize the overlap volume and list of edge sub-volumes

2: \( \text{for each } \Omega_P \in L_{ij} \text{ do} \)

3: \( V_\xi = \delta \Omega_P / n^3; \) \( \triangleright \) volume of sub-volume (\( \xi \)) based on volume of \( \Omega_P \)

4: \( \text{if } \Omega_P \in \partial B_i \text{ or } \Omega_P \in \partial B_j \text{ then} \)

5: \( \text{decompose } \Omega_P \text{ into } n^3 \text{ sub-volumes } \Xi; \)

6: \( \text{isChecked}[k] \rightarrow \text{isInBody}_i, \text{isInBody}_j = [\text{False}] \times n^3; \) \( \triangleright \) create boolean lists

7: \( \phi = [ ]; \) \( \triangleright \) create empty list of sub-volumes to be checked by octree

8: \( \xi_r = \Xi[r]; \) \( \triangleright \) \( r \) is an index of any sub-volume common for both bodies

9: \( \phi = \phi \cup \{N\}_{\xi_k}; \) \( \triangleright \) append neighbour sub-volumes of \( \xi_r \) to \( \phi \)

10: \( \text{while } \phi \text{ is not empty do} \)

11: \( \xi_k = \phi[0]; \) \( \triangleright k \) is the sub-volume index

12: \( \text{if not isChecked}[k] \text{ then} \)

13: \( \text{isChecked}[k] = \text{True}; \)

14: \( \text{isInBody}_i[k] = \text{True}; \)

15: \( \text{isInBody}_j[k] = \text{True}; \)

16: \( \text{if } \xi_k \in B_i \text{ then} \)

17: \( \text{isInBody}_i[k] = \text{True}; \)

18: \( \text{if isInBody}_i[k] \text{ and isInBody}_j[k] \text{ then} \)

19: \( V_{ij}^o = V_{ij}^o + V_{\xi}; \)

20: \( \phi = \phi \cup \{N\}_{\xi_k}; \) \( \triangleright \) append neighbour sub-volumes of \( \xi_k \) to \( \phi \)

21: \( \phi = \phi \setminus \xi_k; \) \( \triangleright \) remove the checked sub-volume from \( \phi \)

22: \( \text{edge sub-volumes detection:} \)

23: \( \text{if } \Omega_P \in E_{ij} \text{ then} \)

24: \( \text{for each } \xi_k \in \Xi \text{ do} \)

25: \( \psi_\ell = \{N\}_{\xi_k}; \) \( \triangleright \) create list of neighbors of the sub-volume

26: \( \text{isNeigIn}_i, \text{isNeigIn}_j = \text{False}; \)

27: \( \text{for each } \psi_\ell \in \Psi \text{ do} \)

28: \( \text{if not isInBody}_i[\ell] \text{ and isInBody}_j[\ell] \text{ then} \)

29: \( \text{isNeigIn}_j = \text{True}; \)

30: \( \text{if isInBody}_i[\ell] \text{ and not isInBody}_j[\ell] \text{ then} \)

31: \( \text{isNeigIn}_i = \text{True}; \)

32: \( \text{if isNeigIn}_i \text{ and isNeigIn}_j \text{ then} \)

33: \( E_{ij}^\xi = E_{ij}^\xi \cup \xi_k; \) \( \triangleright \) append \( \xi_k \) to the list of edge sub-volumes \( E_{ij}^\xi \)

34: \( \text{else} \)

35: \( V_{ij}^o = V_{ij}^o + n^3 \cdot V_{\xi}; \)

return \((V_{ij}^o, E_{ij}^\xi)\)

estimate the cell fraction contributing to \( V_{ij}^o \). The complete \( V_{ij}^o \) is computed by iterating over the whole \( L_{ij} \). The efficiency of the identification of \( \xi \) common for both bodies is improved utilizing the octree algorithm, which allows to limit the number of computationally expensive queries on location of \( \xi \) centroid with respect to \( B_i \) and \( B_j \), see lines 7-21 of Alg. 1.

**Determination of the contact normal** The remaining task is to determine the contact normal \( n \). For \( \Omega^o \)-based computations, the contact normal is evaluated based on the plane defined by the edge cells \( E_{ij} \). However, utilizing the virtual mesh, a finer mesh resolution is available via the sub-volumes \( \Xi \). Thus, identifying edge sub-volumes \( E_{ij}^\xi \) allows for improved contact plane and \( n \).
3 Numerical Experiments

In the following, we first verify the virtual mesh concept on a simple sphere-sphere collision. Afterwards, the ability of the approach to efficiently resolve contact between two general solids is illustrated.

3.1 Approach verification

The proposed virtual mesh algorithm was verified on a comparison of normal contact force $f_{cn}$ evolution during a contact of two spheres. The $f_{cn}$ evolution was compared for a DEM approach leveraging the geometrical simplicity of spheres (sphere DEM) an $\Omega^h$-based computation on a static mesh (original model), new virtual mesh (virtual mesh), and an $\Omega^h$-based computation on an mesh adaptively refined in order to provide the same contact resolution as the virtual mesh algorithm (dynamic mesh).

The test itself is designed as an elastic collision between two spheres. The bottom sphere is fixed at the same position during the whole test, while the top sphere moves towards it with the initial velocity of $u_i = 1 \text{ m/s}$, see Fig. 3a. Both the spheres have the same material properties corresponding to a rubber, i.e. $Y = 100 \text{ MPa}$, $\nu = 0.5$, and $\rho = 1000 \text{ kg/m}^3$. The base CFD mesh $\Omega^h$ resolution corresponded to 20 FV cells per the sphere diameter $d_s$. The $\Omega^h$ itself is a hexahedr of dimensions $3d_s \times 3d_s \times 30d_s$. The initial distance between the sphere centers was $3/2 d_s$.

Examining the test results for the four tested models given in Fig. 3, the following may be devised. First, the sphere DEM, which is comparable to e.g. LIGGGHTS[11] implementation, provides smooth $f_{cn}$ evolution. Furthermore, it is the most computationally efficient approach as no projection of the solids on $\Omega^h$ is not needed. Next, the original model is roughly 5 times less computationally efficient then the sphere DEM. At the same time, it provides only piece-wise continuous $f_{cn}$, which is caused by the too coarse $\Omega^h$ resolution. Finally, both the virtual mesh and dynamic mesh approaches are able to provide results with a similar accuracy and in a good agreement with the sphere DEM model. However, while virtual mesh is approximately 7 times less efficient than the sphere DEM approach and by 30% less efficient than the original model, the dynamic mesh approach requires 236 times more time than the sphere DEM.

Figure 3: The evolution of the normal contact force during contact for the four compared DEM models, (a) outline of the test, (b) sub-volumes used in the virtual mesh decomposition, (c) particles in the contact.
Figure 4: Analysis of the contact test for arbitrarily-shaped bodies, (a) normal contact force and (b) relative error between methods during the contact, (c) green body center of mass trajectory, and relative position of bodies (d) at $t = 0$, (e) in the contact, (f) at $t = t_{end}$.

### 3.2 Collision of the arbitrary bodies

The advantageous properties of the virtual mesh are illustrated on a solution of contact between to arbitrarily shaped. In this test, the sphere DEM approach is not applicable at all and the original model, i.e. $\Omega^h$-based computation without the adaptive mesh refinement and the same $\Omega^h$ resolution as in the previous test, provides completely nonphysical results. Consequently, the test was limited to a comparison of the virtual mesh approach to the dynamic mesh one.

Once more, the test comprised two rubber particles. The bottom particle was fixed and the top one moved to it with the same initial velocity and from the same distance as in the previous test. Furthermore, the same $\Omega^h$ was used. The test results are given in Fig. 4. The $f_{c,n}$ evolution shown in Fig. 4a seems to be in a perfect agreement for the two approaches. However, focusing on the actual moving particle trajectory, which is visualized in Fig. 4c, there are some discrepancies between the approaches. The difference in the moving particle centroid position initiate at the moment of contact and increases up to $1\%$ towards the test end, see Fig. 4b. Still, this behavior is to be expected as the particles trajectories are Lyapunov-unstable, i.e. any small difference in $f_{c,n}$ during the contact solution leads to qualitatively different trajectory. Finally, note the computational efforts necessary to solve the contact. The dynamic mesh approach is more than 10 times less efficient than the virtual mesh. Furthermore, increasing the Young modulus of the particles in contact leads to decrease in $V_{ij}^m$ magnitudes and increases the requirements on the DEM mesh resolution. Consequently, to resolve a contact between e.g. steel particles with $Y \approx 210\,\text{GPa}$ via the dynamic mesh approach is computationally unfeasible.
4 Conclusion

In this paper, we proposed the virtual mesh extension to our CFD-DEM solver allowing for efficient DEM solution of contact between arbitrarily shaped solids projected onto a CFD mesh. Utilizing a contact between two spheres, the virtual mesh algorithm was verified against different DEM approaches, including a one that leverages the simplicity of the sphere geometry and is completely independent on the CFD mesh. However, in general, the CFD mesh resolution plays a crucial role in computation of contact forces for irregular bodies. The proposed algorithm is able to provide the same contact solution accuracy as an adaptive refinement of the CFD mesh while requiring only a fraction of the computational resources. Enhancing our CFD-DEM solver by the virtual mesh concept extends its capabilities towards simulations involving a transport of irregular bodies dispersed in a fluid, e.g. suspensions.

Acknowledgment

The work was financially supported by the Czech Science Foundation (GA19-22173S), by the institutional support RVO:61388998, and by the Centre of Excellence for Nonlinear Dynamic Behaviour of Advanced Materials in Engineering CZ.02.1.01/0.0/0.0/15_003/0000493 (Excellent Research Teams) in the framework of Operational Programme Research, Development and Education. Martin Kotouč Šourek thanks Johnson Matthey Technology Centre in Sonning Common for the support and feedback.

References