Title

Coupled three-dimensional discrete element-lattice Boltzmann methods for fluid-solid interaction with polyhedral particles

Journal

International Journal for Numerical and Analytical Methods in Geomechanics, 43(14)

ISSN

0363-9061

Authors

Gardner, Michael
Sitar, Nicholas

Publication Date

2019-10-10

DOI

10.1002/nag.2972

Peer reviewed

Coupled three-dimensional discrete element-lattice Boltzmann methods for fluid-solid interaction with polyhedral particles

Michael Gardner[1] Nicholas Sitar[2]

[1]NHERI SimCenter, University of California, Berkeley, California, U.S.A.
[2]Department of Civil and Environmental Engineering, University of California, Berkeley, California, U.S.A.

Correspondence Michael Gardner, NHERI SimCenter, University of California, Berkeley, California, 94720, U.S.A. Email: mhgardner@berkeley.edu

Present Address Richmond Field Station, Building 454, Room 126, 1301 S 46th Street, Richmond, California, U.S.A.

Summary

Interaction between solid particles and fluid is of fundamental interest to scientists and engineers in many different applications—cardiopulmonary flows, aircraft and automobile aerodynamics, and wind loading on buildings to name a few. In geomechanics, particle shape significantly affects both particle-particle and particle-fluid interaction. Herein, we present a generalized method for modeling the interaction of arbitrarily shaped polyhedral particles and particle assemblages with fluid using a coupled discrete element method (DEM) and lattice Boltzmann method (LBM) formulation. The coupling between DEM and LBM is achieved through a new algorithm based on a volume-fraction approach to consider three-dimensional convex polyhedral particles moving through fluid. The algorithm establishes the interaction using linear programming and simplex integration and is validated against experimental data. This approach to modeling the interaction between complex polyhedral particles and fluid is shown to be accurate for directly simulating hydrodynamic forces on the particles.

KEYWORDS: discrete element method, fluid-solid interaction, lattice Boltzmann method, linear programming, polyhedral particles, simplex integration

1 INTRODUCTION

Interaction between solid particles and fluid is of fundamental interest to scientists and engineers in many different applications—cardiopulmonary flows, aircraft and automobile aerodynamics, and wind loading on buildings to name a few. In geomechanics, fluid-solid interaction factors into almost every type of analysis and it could be argued that the presence of water is the most important effect to capture correctly. Suspended particle transport, rock scour, soil liquefaction, and effective stress analyses are some of the

most pertinent examples of fluid-solid interaction that in most cases require dynamic analyses. For these types of dynamic analyses, the hydrodynamic forces and moments exerted on the solid particles need to be accounted for when integrating the equations of motion for the solid phase. Also, the effect of the solids in and moving through the fluid needs to be incorporated into the fluid solver. Simulations that capture this interaction can follow several approaches: a locally-averaged interaction between the two phases,[1-5] computational homogenization,[6-8] or direct simulation of hydrodynamic forces on the solid particles.[9-13] In the locally-averaged approach, the fluid-solid coupling is done by averaging the interactions over a representative volume, and all particles within a local region experience the same hydrodynamic forces. This makes the method less computationally expensive compared with direct simulation, since the number of solid particles is greater than the number of fluid cells. While appropriate in certain applications, it does not offer sufficient resolution when trying to establish the hydrodynamic interaction for individual particles. Direct simulation of the fluid-solid interaction attempts to overcome this shortcoming by having a much higher density fluid mesh compared with the number of solid particles. This approach is able to capture the variation in hydrodynamic forces on individual particles, but it does come at a much higher computational cost. Computational homogenization bridges the gap between the locally-averaged approach and direct simulation by considering the coupling directly only at certain locations in the solution within a representative volume.

When modeling the solid phase, the shape of the particles often plays an important role in its mechanical response. For example, the interlocking of irregularly shaped sand particles is known to give higher strengths compared with more rounded particles, and the kinematic response of rock slopes is governed by the orientation of the rock blocks relative to slope geometry. Therefore, the numerical method used to model the solid phase must be capable of capturing the particle shapes and the discrete, particle-to-particle interactions. The discrete element method (DEM)[14-16] and discontinuous deformation analysis (DDA)[17, 18] have gained popularity within geomechanics to model the particulate nature of geomaterials. DDA parallels the finite element method (FEM) in that the blocks are described through a system of equations where each element is a distinct, isolated block. For dynamic analyses, the system of equations is generally solved implicitly though it is possible to integrate them explicitly. DEM, on the other hand, considers the motion of each particle individually using explicit time integration. The localized nature of DEM—in the event that there are no long-range interaction forces—and its relatively straightforward extension to three

dimensions makes it an attractive candidate for modeling particle assemblages and for coupling the solid phase model with a fluid solver.

In terms of simulating the fluid phase, the most established methods in computational fluid dynamics (CFD), such as the FEM[19] and finite volume method (FVM),[20] are able to capture complex boundary shapes through irregular and unstructured grids and, in the case of FEM, are amenable to higher order methods. However, the mesh generation can be quite complicated and computationally intensive which can be prohibitive if there are solid particles moving through the fluid domain—generally necessitating remeshing at every time step. A comparatively newer method for CFD is the lattice Boltzmann method (LBM).[21, 22] What makes LBM particularly attractive for modeling fluid-solid interaction is the localized nature of the method and, more importantly, the relative ease with which complex shapes moving through the fluid domain can be accommodated. As a solid particle moves through the fluid domain, the state of the nodes that the solid interacts with is updated. On the basis of the status of the node, the presence of the solid is accounted for in the fluid solution and the effect of the fluid on the solid is also considered. The change in the status of each node is incorporated in the computations locally.

For assemblages of arbitrarily shaped polyhedral particles or blocks, the individual behavior of solid particles is important for capturing its mechanical response. In such cases, direct simulation of solid-fluid interaction becomes necessary to accurately capture the system's behavior. The coupling process between DEM and LBM when solids are allowed to move through the fluid mesh is comparatively simpler and less computationally expensive than other CFD methods. A significant body of work considering coupling between spherical particles—both single and clustered spheres—and sphero-polyhedral particles and fluid using DEM and LBM exists.[11, 13, 23-30] However, some of these methods are either not capable of considering more complex polyhedral shapes or introduce approximations of the particle shape during the coupling process. Others implement bounce-back–based coupling algorithms that are not exactly mass conserving or that require interpolations, which may be troublesome in dense particle assemblies with small pore sizes.[31] To this end, a coupling algorithm was developed based on the volume-fraction approach[9] to consider three-dimensional convex polyhedral particles moving through the fluid mesh where the solid particles are modeled using DEM and the fluid phase is modeled using LBM. The coupling process is implemented using the multi-relaxation-time (MRT) LBM which offers improved numerical stability and accuracy.[32] Overlap between the solid DEM particles and fluid cells in the LBM is established through a constrained optimization problem. In the event of overlap, the volume-

fraction of solid in a fluid cell is calculated analytically using simplex integration. The new coupling algorithm—which synthesizes MRT LBM, constrained optimization, and simplex integration—is validated against experimental data.

2 DISCRETE ELEMENT METHOD

The DEM is able to simulate the behavior of discontinuous systems composed of many particles by describing the motion of each individual particle and its interaction with its neighbors. The motion of the particle is described by Newton's second law, while the interaction of the particle with its neighbors is captured through explicitly considering particle-particle forces based on a contact law. The two main phases in DEM computations are contact detection and contact interaction calculations. The contact detection phase establishes which particles are in contact, while the interaction calculations evaluate the forces and moments between the contacting particles.

2.1 Contact detection

The contact detection phase is the most computationally expensive portion of DEM simulations, accounting for approximately 80% of the total simulation time.[33] Two steps comprise contact detection: neighbor search and contact resolution. During the neighbor search, a particle's nearest neighbors—particles that are close enough to possibly be in contact within a given time period or step—are identified. Neighbor search algorithms generally implement either tree-based searching or spatial binning where tree-based algorithms are $O(NIn(N))$ and spatial binning algorithms are $O(N)$. [34] For this research, the CGRID spatial binning algorithm[34] was used.

The neighbor search determines which particles are “close enough” such that they need to be checked for possible contact in the contact resolution step. In contact resolution, the neighboring particles' bounding spheres are first checked for overlap. If they do, further contact resolution is required. If they do not, the particles are not in contact. In the case where further contact resolution is required, the contact detection problem is recast as a convex optimization.[35] Only a brief overview of this approach is provided, as a full description is given in Boon et al.[35]

The polyhedral particle shape, as shown in Figure 1, is defined completely by the N planes that bound it:

$$\boldsymbol{a}_i^T \boldsymbol{x} - d_i \leq 0, \quad i = 1, \ldots, N \tag{1}$$

where a_i represents the normal vector to the ith plane bounding the particle and d_i is the distance of that plane from some local origin. Contact between

two polyhedral particles is then established by solving the following linear program:

$$\text{minimize } s$$
$$\boldsymbol{a}_i^T \boldsymbol{x} - d_i \leq s, \quad i = 1, \ldots, N_A + N_B \tag{2}$$

where N_A and N_B are the number of planes of the two neighboring particles. The two particles are in contact if $s<-\epsilon$ where ϵ is a specified numerical tolerance. If the particles are in contact, the contact point is taken as the analytic center of the region of overlap between the two contacting particles. The analytic center, shown in Figure 1, is calculated using the log-barrier method with Newton's method. The contact point is used to calculate the contact normal and overlap and is assumed to be the location where the contact force is applied to the contacting particles.

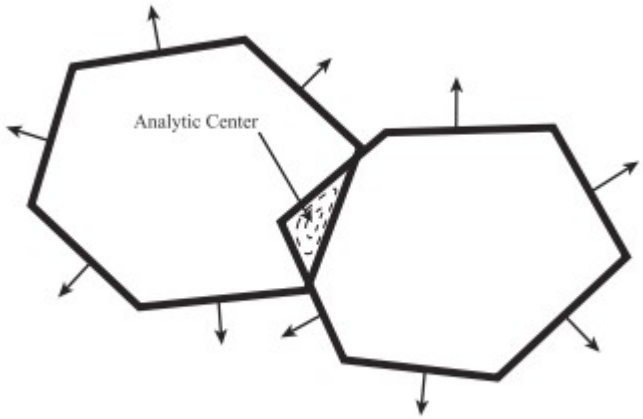


Figure 1. Two colliding particles with analytic center taken as contact point. Arrows indicate the direction of normal vectors to particle faces (modified, based on Boon et al[35])

2.2 Contact forces and moments

Once contact between two particles has been established, the next step is to describe how they interact with each other—what are the forces and moments between the two particles. The contact normal and overlap calculated in the contact resolution phase serve as inputs to establish the interaction forces between the particles. In its basic formulation, DEM is modular in terms of how the contact forces are described and many different formulations exist for calculating contact forces. In the simplest case, the contact between two particles is described as linear elastic in the contact normal direction and frictional, using Coulomb friction with cohesion, in the tangential direction.[16] More involved contact models for spherical particles are based on elasticity theory[36-38] while numerous models have been proposed for polyhedral rock rock blocks.[39-43] Overall, the underlying

implementation of the DEM formulation remains unchanged and the contact model takes the results from the contact resolution phase as inputs.

Once the forces and moments due to inter-particle contacts have been calculated, the particle positions are updated. The equations of motion for an individual particle are:

$$\ddot{\boldsymbol{x}}_i + \alpha \dot{\boldsymbol{x}}_i = \frac{\boldsymbol{F}_i}{m_i} + \boldsymbol{g}_i$$
$$\dot{\boldsymbol{\omega}}_i + \alpha \boldsymbol{\omega}_i = \frac{\boldsymbol{M}_i}{I_i} \quad (3)$$

where $\ddot{\boldsymbol{x}}_i$ and $\dot{\boldsymbol{\omega}}_i$ are the translational and rotational acceleration of particle i; F_i and M_i are the total force and moment acting on particle i; α is a damping constant that can be set independently for translation and rotation; m_i and I_i are the mass and moment of inertia of particle i; and g_i is the gravitational acceleration. The particle translational motion is integrated using a velocity Verlet finite difference approach,[44] while the rotational motion is updated using a quaternion-based fourth-order Runga-Kutta approach.[45]

3 LATTICE BOLTZMANN METHOD

The fluid phase is modeled using the LBM. LBM arrives at the solution for various problems in fluid dynamics by solving a discrete form of the Boltzmann equation—the so-called lattice Boltzmann equation 21:

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i \Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) + \Omega_i(\boldsymbol{x}, t), \quad (4)$$

where c_i is the discrete set of velocities which limits the continuous particle velocity to a carefully selected subset. In this research, the D3Q27 velocity set[46] as shown in Figure 2 was implemented. The D3Q27 velocity set is more expensive in terms of computations and memory compared with other velocity sets such as the D3Q19 or D3Q15 velocity sets; however, the D3Q19 and D3Q15 velocity sets have been shown to produce unphysical secondary currents at high Reynolds numbers and violate Galilean invariance [47, 48] for certain flow configurations and boundary conditions. Given these limitations, the D3Q27 velocity set was selected for its robustness in a wider range of flow conditions. The discrete velocities and their accompanying weights for the D3Q27 velocity set are shown in Table 1.

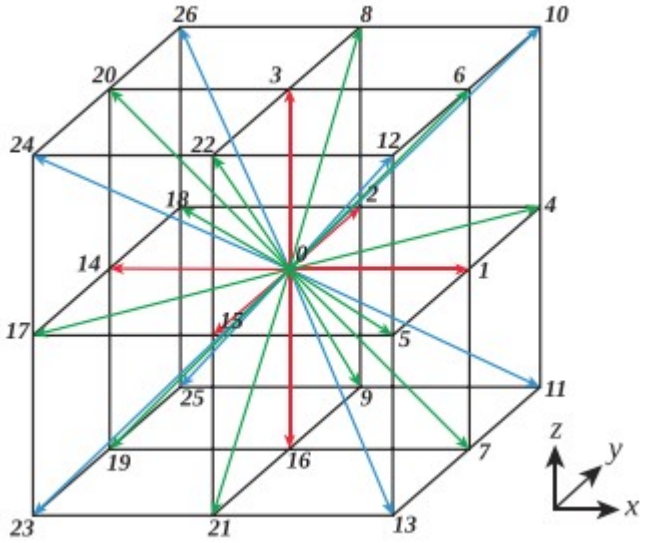


Figure 2. D3Q27 velocity set showing discrete velocity ordering

TABLE 1 D3Q27 velocity set

Velocities c_i	Number	Weight w_i
$(0,0,0)$	1	8/27
$(\pm1,0,0),(0,\pm1,0),(0,0,\pm1)$	6	2/27
$(\pm1,\pm1,0),(\pm1,0,\pm1),(0,\pm1,\pm1)$	12	1/54
$(\pm1,\pm1,\pm1)$	8	1/216

Equation 4 describes particles $f(x_i,t)$ moving with velocity c_i to a neighboring point located at $x+c_it$. This is known as the *streaming* step. Additionally, the collision operator Ω_i redistributes particles among the populations f_i at each point—this redistribution of particles models particle collisions. This is known as the *collision* step. Together the streaming and collision steps are the fundamental concept of the lattice Boltzmann equation. The basic variable in LBM, as shown in Equation 4, is the discrete-velocity distribution function, $f_i(x,t)$. The distribution function represents the density of particles with velocity c_i at time t and position x. The macroscopic fluid mass density and momentum are calculated through weighted sums in velocity space, known as moments, of f_i:

$$\rho(\boldsymbol{x},t)=\sum_i f(\boldsymbol{x},t)$$
$$\rho\boldsymbol{u}(\boldsymbol{x},t)=\sum_i \boldsymbol{c}_i f(\boldsymbol{x},t) \quad (5)$$

The collision operator, Ω_i, models inter-particle collisions by redistributing particles among different populations f_i at each point. The most common and simple collision operator is the Bhatnagar-Gross-Krook (BGK) collision operator[49]:

$$\Omega_i(\boldsymbol{x},t) = -\frac{f_i(\boldsymbol{x},t) - f_i^{eq}(\boldsymbol{x},t)}{\tau}. \quad (6)$$

This equation implies that all populations f_i decay, or relax, to their equilibrium state f_i^{eq} at the same rate τ. The discrete form of the equilibrium distribution function f_i^{eq} is[50]:

$$f_i^{eq}(\boldsymbol{x},t) = \omega_i \rho \left[1 + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u})^2}{2c_s^4} - \frac{\|\boldsymbol{u}\|^2}{2c_s^2}\right]. \quad (7)$$

where u and ρ are the fluid velocity and density; c_i and w_i are the discrete velocity and weight from the velocity set; and c_s is the speed of sound. For the D3Q27 lattice, $c_s = \Delta x / \left(\sqrt{3}\Delta t\right)$. However, the simplicity of the BGK collision operator comes at a cost: reduced accuracy, particularly at large viscosities, and stability, particularly at small viscosities.[31] To overcome these shortcomings, the multi-relaxation-time (MRT) collision operator[32] is useful since each of the moments can be relaxed at different time scales to achieve better stability and accuracy. In order to do this, all populations f_i must first be transformed to moment space, in which the collision step is performed, and then transformed back into population space, where the streaming step is performed. The MRT form of the lattice Boltzmann equation is then:

$$|f(\boldsymbol{x} + \boldsymbol{c}_i t, t + \Delta t)\rangle - |f(\boldsymbol{x},t)\rangle = -\boldsymbol{M}^{-1}\hat{\boldsymbol{S}}\boldsymbol{M}\left[|f(\boldsymbol{x},t)\rangle - |f^{eq}(\boldsymbol{x},t)\rangle\right], \quad (8)$$

where M is the transformation matrix that transforms the distribution functions from velocity space to moment space; and $\hat{\boldsymbol{S}}$ is the diagonal collision matrix: $\hat{\boldsymbol{S}} = diag\left(s_0, s_1, \ldots, s_{q-1}\right)$. The values along the diagonal of $\hat{\boldsymbol{S}}$ are relaxation parameters for the different moments.

The transformation matrix M based on the ordering of the D3Q27 lattice used in this research was calculated using the orthogonal moment set from Geier et al.[51] Appendix APPENDIX A1 gives the calculated values for M and M^{-1}. The relaxation parameters along the diagonal of $\hat{\boldsymbol{S}}$ for the D3Q27 lattice were set to the optimized values proposed by Suga et al[46]:

$$\hat{\boldsymbol{S}} = diag(0,0,0,0,s_4,s_5,s_5,s_7,s_7,s_7,s_{10},s_{10},s_{10},s_{13},s_{13},s_{13},$$
$$s_{16},s_{17},s_{18},s_{18},s_{20},s_{20},s_{20},s_{23},s_{23},s_{23},s_{26}) \quad (9)$$

where:

$$s_4 = 1.54, \quad s_{10} = 1.5, \quad s_{13} = 1.83, \quad s_{16} = 1.4,$$
$$s_{17} = 1.61, \quad s_{18} = s_{20} = 1.98, \quad s_{23} = s_{26} = 1.74. \quad (10)$$

The macroscopic Navier-Stokes behavior is recovered when the kinematic shear viscosity ν is related to the corresponding components of $\hat{\boldsymbol{S}}$:

$$\nu = c_s^2 \left(\frac{1}{s_5} - \frac{1}{2} \right) \Delta t = c_s^2 \left(\frac{1}{s_7} - \frac{1}{2} \right) \Delta t. \quad (11)$$

The inclusion of body forces in LBM manifests itself as an additional source term, S_i:

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i t, t + \Delta t) - f_i(\boldsymbol{x}, t) = \Omega_i(\boldsymbol{x}, t) + S_i(\boldsymbol{x}, t). \quad (12)$$

The forcing scheme proposed by Guo et al[52] was implemented in this research. Following this scheme, the equilibrium and macroscopic fluid velocity are defined as:

$$\boldsymbol{u}^{eq} = \boldsymbol{u} = \frac{1}{\rho} \sum_i f_i \boldsymbol{c}_i + \frac{\boldsymbol{F} \Delta t}{2\rho}, \quad (13)$$

where F is the force density; $F=\rho g$ in the case of a gravitational force. The forcing source term takes the following form:

$$S_i = \left(1 - \frac{\Delta t}{2\tau} \right) w_i \left(\frac{\boldsymbol{c}_i - \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}) \boldsymbol{c}_i}{c_s^4} \right) \cdot \boldsymbol{F}, \quad (14)$$

where τ is the BGK relaxation time. The inclusion of relaxation parameters based on an MRT collision operator was performed based on Li et al.[53]

4 FLUID-SOLID COUPLING

Describing the interaction between the fluid and solid phases requires the two separate models, DEM and LBM, to exchange information. The presence of solids in the fluid mesh has to be factored into the fluid response, while the hydrodynamic forces and moments acting on the solids need to be included in the equations of motion for the individual particles. The coupling process is achieved though the introduction of a special boundary condition based on the volume-fraction of solid in fluid cells. This allows the polyhedral particles to move through the fluid mesh while still maintaining a similar form of the lattice Boltzmann equation.

First introduced by Noble and Torczynski,[9] the partially saturated method (which we refer to as the volume-fraction approach to avoid confusion with partially saturated soil mechanics) accounts for the presence of complex shaped solids within the fluid mesh by considering the volumetric solid content of each of the lattice cells. As a particle moves through the fluid mesh, it may partially or completely cover fluid cells, as shown in Figure 3. The LBE with the BGK collision operator is modified to accommodate the solid phase by introducing an additional solid collision operator:

$$f_i(\boldsymbol{x}+\boldsymbol{c}_i\Delta t, t+\Delta t) = f_i(\boldsymbol{x}, t) + \left[1 - \sum_s B(\epsilon_s, \tau)\right]\Omega_i^{BGK} + \sum_s B(\epsilon_s, \tau)\Omega_i^s. \quad (15)$$

where ϵ_s is the volumetric solid fraction for each particle intersecting the fluid node and $B(\epsilon_s,\tau)$ is a weighting function. $B(\epsilon_s,\tau)$ ranges from 0 (pure fluid) to 1 (pure solid). When $B(\epsilon_s,\tau)=0$, the standard lattice Boltzmann equation is recovered, while of $B(\epsilon_s,\tau)=1$, only the solid collision operator participates in the collision step. The collision operator for solid nodes is:

$$\Omega_i^s = f_{-i}(\boldsymbol{x}, t) - f_i(\boldsymbol{x}, t) + f_i^{eq}(\rho, \boldsymbol{u}_s) - f_i^{eq}(\rho, \boldsymbol{u}), \quad (16)$$

where u_s is the velocity of the solid particle at time $t+\Delta t$ at the fluid node. This form of Ω_i^s is based on the bounce-back of the non-equilibrium portion of the particle distributions.[54] The weighting funciton is expressed as:

$$B(\epsilon_s, \tau) = \frac{\epsilon_s(\tau/\Delta t - 1/2)}{(1+\epsilon_s) + (\tau/\Delta t - 1/2)}. \quad (17)$$

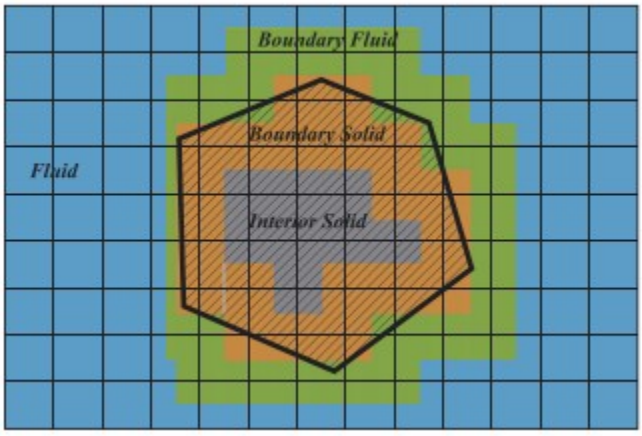


Figure 3. Polyhedral particle moving through fluid mesh. Background grid indicates lattice nodes which are at the center of fluid cells. Fluid cells are pure fluid, interior solid cells are pure solid. Boundary fluid and boundary solid cells are some proportion of fluid and solid

The hydrodynamic force and torque acting on a particle moving through the fluid mesh is calculated as:

$$\boldsymbol{F} = \frac{\Delta x^3}{\Delta t}\sum_{x_n} B(\boldsymbol{x_n})\left(\sum_i \Omega_i^s \boldsymbol{c}_i\right)$$

$$\boldsymbol{T} = \frac{\Delta x^3}{\Delta t}\sum_{x_n}\left(B(\boldsymbol{x}_n)(\boldsymbol{x}_n - \boldsymbol{x}_{CM}) \times \sum_i \Omega_i^s \boldsymbol{c}_i\right), \quad (18)$$

where x_n are all the lattice nodes that are interacting with the particle and x_{CM} is the location of the center of mass of the particle. The

summation i runs over all directions of the particular lattice velocity set in use— 27 in the case of this research.

4.1 Volumetric solid fraction

The behavior of the fluid-solid interaction in the case when a fluid cell is neither pure fluid or pure solid—boundary fluid or boundary solid cells as shown in Figure 3—is dictated by the weighting factor B. The value of the weighing function is influenced by the collision operator parameters and, to a greater extent, the volumetric solid content of the cell. Therefore, it is important that the volumetric solid content be calculated as accurately and efficiently as possible. In terms of calculating the volumetric solid fraction, current coupling algorithms approximate the shape of the particles in terms of the shape of the region of overlap—Owen et al[11] for spherical particles and Galindo-Torres[13] for sphero-polyhedra. For this research, a new method was developed to calculate the volumetric solid content analytically for convex polyhedra to determine the value of the weighting function. This new method was applied using an MRT collision operator, though it can be used with the BGK or two-relaxation-time collision operators.

The first step in calculating the volumetric solid content is to determine whether the particle and fluid cell overlap, as shown in Figure 4. This is essentially the same problem as establishing contact between two polyhedral particles. Thus, for this phase of the calculations, the methods used in contact detection for DEM can also be applied to establish whether a particle intrudes on a fluid cell. Here, a linear programming approach analogous to that of Boon et al[35] as shown in Equation 2 is used to establish overlap between the fluid cell and particle:

$$\begin{aligned} &\text{minimize s} \\ &\boldsymbol{a}_i^T \boldsymbol{x} - d_i \leq s, \quad i = 1, \ldots, N_s + N_F \end{aligned} \tag{19}$$

where N_s is the number of faces that define the particle and N_F is the number of faces that define the fluid cell—four faces for a square in two dimensions and size faces for a cube in three dimensions. The particle and fluid cell overlap if $s<-\epsilon$ where ϵ is a specified numerical tolerance. If the particle and fluid cell do not overlap then the cell is pure fluid with a volumetric solid content of zero. However, if the particle and fluid cell do overlap, it is necessary to perform further computations to determine the volumetric solid content.

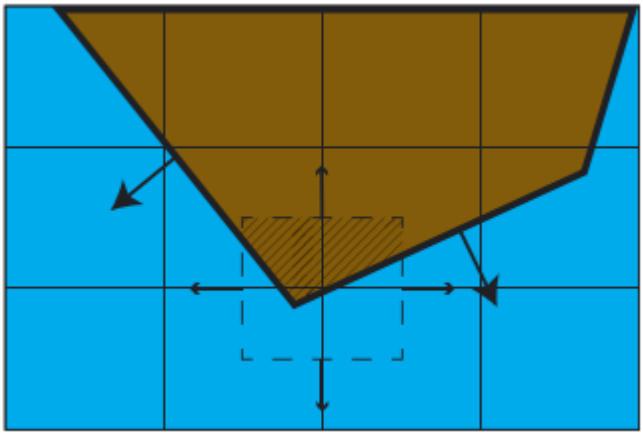


Figure 4. Closeup of polyhedral particle overlapping fluid cell. The fluid cell is described by the normals to the cell faces and their distance from the lattice node at the center of the cell. The hatched region is where the particle and fluid cell overlap—this is the solid content of the fluid cell

The region of overlap between the particle and fluid cell shown for the two-dimensional case in Figure 4 comprises the solid content. This region is described by a subset of the faces that define the particle and fluid cell; with this subset it is possible to calculate the solid content. The minimal set of faces that describe this region of overlap is established by checking all faces of the particle and fluid cell for redundancy. A particular face $n^T x \leq d$ from the set N_s+N_F is checked for redundancy by solving the linear program:

$$\begin{aligned} &\text{maximize} \quad \boldsymbol{n}^T \boldsymbol{x} \\ &\boldsymbol{a}_i^T \boldsymbol{x} \leq d_i, \quad i = 1, \ldots, N_s + N_F \end{aligned} \tag{20}$$

[Correction added on 14 August 2019, after first online publication: equation 20 has been corrected.]

The face is not redundant if $|\boldsymbol{n}^t \boldsymbol{x} - d| < \epsilon$. This approach is similar to the removal of redundant faces during fractured rock mass generation as described in Boon et al[55]; however, other methods can be used for removing redundant faces[56, 57] as they are simply linear constraints.

With this minimal set, it is now possible to calculate the volumetric solid fraction in the fluid cell. The volume of the region bounded by the this minimal set is calculated using simplex integration[58]:

$$V_{\text{overlap}} = \int\int\int_V dx\,dy\,dz$$
$$= \sum_{i=1}^{s}\sum_{k=1}^{n(i)} S_{P_0 P_1^i P_k^i P_{k+1}^i}(0,0,0)$$
$$= \frac{1}{6}\sum_{i=1}^{s}\sum_{k=1}^{n(i)} \begin{vmatrix} x_1^i & y_1^i & z_1^i \\ x_k^i & y_k^i & z_k^i \\ x_{k+1}^i & y_{k+1}^i & z_{k+1}^i \end{vmatrix} \quad (21)$$

Equation 21 describes the summation of the volumes of tetrahedra S_i—a tetrahedron is a three-dimensional simplex—that together form the three-dimensional particle. This assumes that the vertices $P_1,\ldots,P_n$ describing each of the faces of the particle are oriented counter-clockwise relative to the outward normal of the face. All vertices are specified relative to a local origin P_0, in this case set to the location of the center of the fluid cell in question.

After, the volume of the region of overlap is calculated, the volumetric solid content is given by $\epsilon_s = V_{\text{overlap}}/V_{\text{cell}}$ where the volume of the cell is the volume of a cube with side length Δx. For the D3Q27 MRT collision operator used in this research, the weighting function is then given by:

$$B = \frac{\epsilon_s\left(\frac{1}{s_5} - \frac{1}{2}\right)}{(1-\epsilon_s) + \left(\frac{1}{s_5} - \frac{1}{2}\right)} = \frac{\epsilon_s\left(\frac{1}{s_7} - \frac{1}{2}\right)}{(1-\epsilon_s) + \left(\frac{1}{s_7} - \frac{1}{2}\right)} \quad (22)$$

5 COUPLING ALGORITHM VALIDATION

The new fluid-solid coupling algorithm was implemented in C++. All of the individual pieces of functionality in the code were unit tested. Beyond simple unit tests, the coupling algorithm was tested for physical correctness by comparing numerical results with several different correlations based on physical experiments.

5.1 Comparison with experimental data

The numerically calculated drag coefficient C_D was compared with regressions on experimental results.[59-61] These results incorporate both shape of the particle and the Reynolds number. The value of C_D was computed for Reynolds numbers of 0.3, 30, 90, and 240 for a cube in uniform flow. The upstream face of the cube was oriented perpendicular to flow. The Reynolds number is given by:

$$Re = \frac{\|\boldsymbol{u}\| d}{\nu} \quad (23)$$

where u is the velocity of the undisturbed fluid and ν is the kinematic viscosity. The characteristic dimension of the the cube d is defined as the diameter of the volume-equivalent sphere. The drag coefficient is defined as:

$$C_D = \frac{F_D}{\frac{1}{2}\rho u^2 A_{\text{proj}}} \quad (24)$$

where F_D is the force component in the direction of flow, ρ is the fluid density, and A_{proj} is the projected frontal area of the volume-equivalent sphere ($A_{\text{proj}}=\pi(d/2)^2$).

As shown in Figure 5, the calculated drag coefficient from the numerical analyses matches all three experimental correlations well over the tested range of Reynolds numbers. The three solid lines are the curves of regressed experimental data and the red squares are the numerical results.

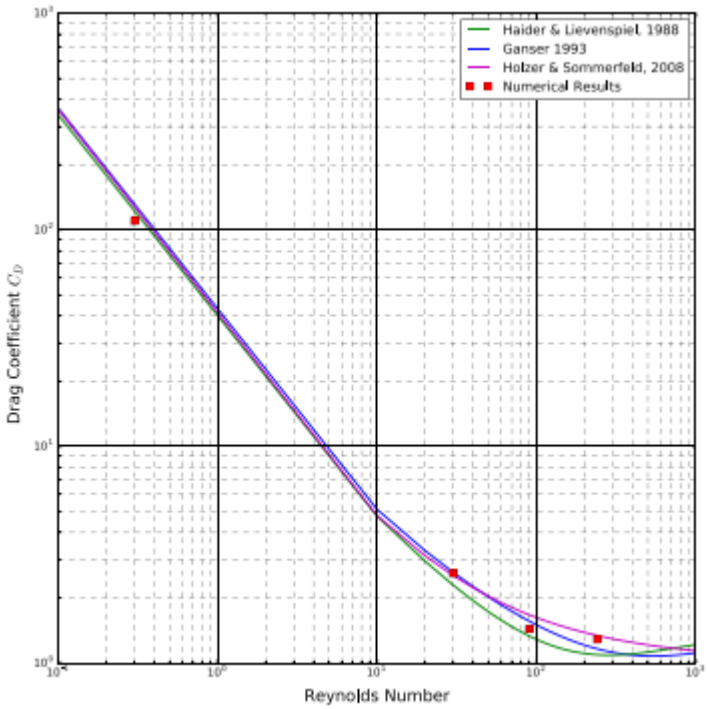


Figure 5. Comparison of numerically calculated drag coefficient C_D for a cube in uniform flow with regressed experimental data.[59-61] The upstream face of the cube is oriented perpendicular to flow

5.2 Cube rotated in uniform flow

Next, the orientation of the cube relative to the flow field was changed to check whether the coupling algorithm is able to capture the change in drag

coefficient due to the change in projected area of the cube relative to the direction of fluid flow. The upstream boundary was set to a constant velocity (u=⟨1.0,0.0,0.0⟩ m/s), while the downstream boundary was a non-reflecting characteristic boundary. All other boundaries were periodic. The cube was rotated around an axis perpendicular to the flow field and simulations were run at rotation angles of 0°, 15°, 30°, 45°, 60°, 75°, and 90° at Reynolds numbers 0.3, 30, 90, and 240, respectively. Figure 6 shows the three-dimensional mesh for the case where the cube was rotated 15°. Figure 7A shows a section through this three-dimensional domain in the near-vicinity of the particle to illustrate the mesh density and Figure 7B shows the node status for the coupling process. Red nodes are fluid, pink nodes are solid, dark blue nodes are boundary fluid, and light blue nodes are boundary solid. The mesh spacing was 0.035 m in all directions giving a mesh with 5 088 448 nodes. The model time was 6.0 seconds with a time step of 0.0005 seconds. This simulation took approximately 15.7 minutes to complete on a machine with two Intel Xeon E5-2630 v2 (2.3 GHz) CPUs with 12 cores each (24 cores total) and 20GB of memory.

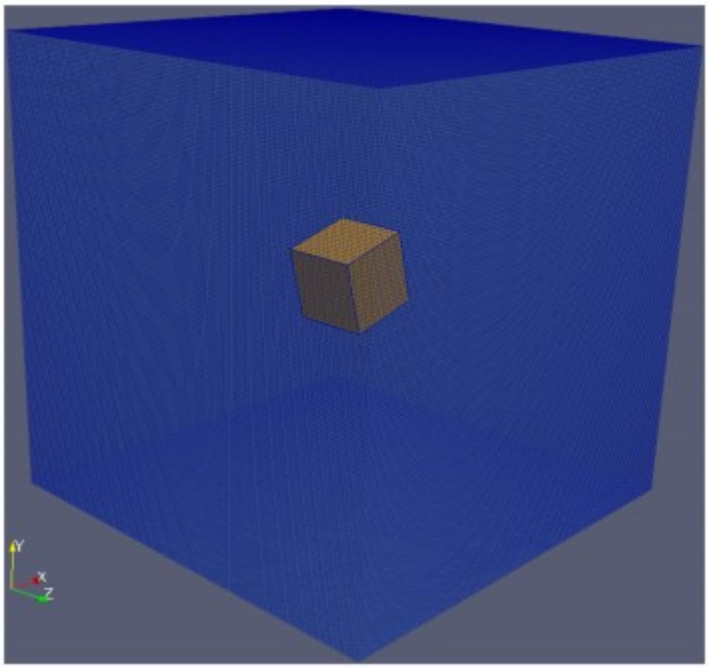


Figure 6. Cube embedded in three-dimensional fluid mesh

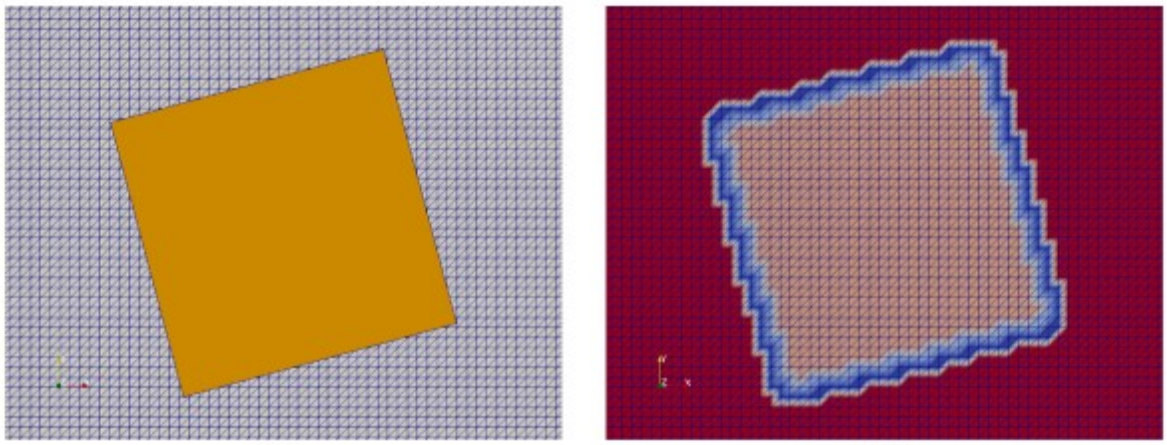


Figure 7. Sections through three-dimensional domain showing mesh density in the vicinity of the cube. A, Cube embedded in fluid mesh. Cube is rotated 15° around axis perpendicular to flow direction. Cube side length is 1 m and mesh spacing is 0.035 m. B, Status of fluid nodes for coupling process. Red nodes are fluid, pink nodes are solid, dark blue nodes are boundary fluid, and light blue nodes are boundary solid

Figure 8 shows the calculated values of C_D for the tested rotation angles θ. The calculated values for the drag coefficient are mirrored around 45° as is expected since the cube is isometric. The maximum projected cross-sectional area is obtained when the cube is rotated 45° relative to the flow direction, giving the highest drag coefficient. Any rotation away from this orientation leads to a lower projected cross-sectional area and hence a lower drag coefficient with the lowest values at 0° and 90°. The coupling algorithm is able to capture this behavior. Additionally, C_D is more sensitive for higher Reynolds numbers which is consistent with observed behavior. Figure 9A shows the resulting velocity field for the case where the cube was rotated 15° relative to flow at Re=240. The velocity magnitude is shown in m/s. Figure 9B shows this same cube with velocity vectors colored based on pressure.

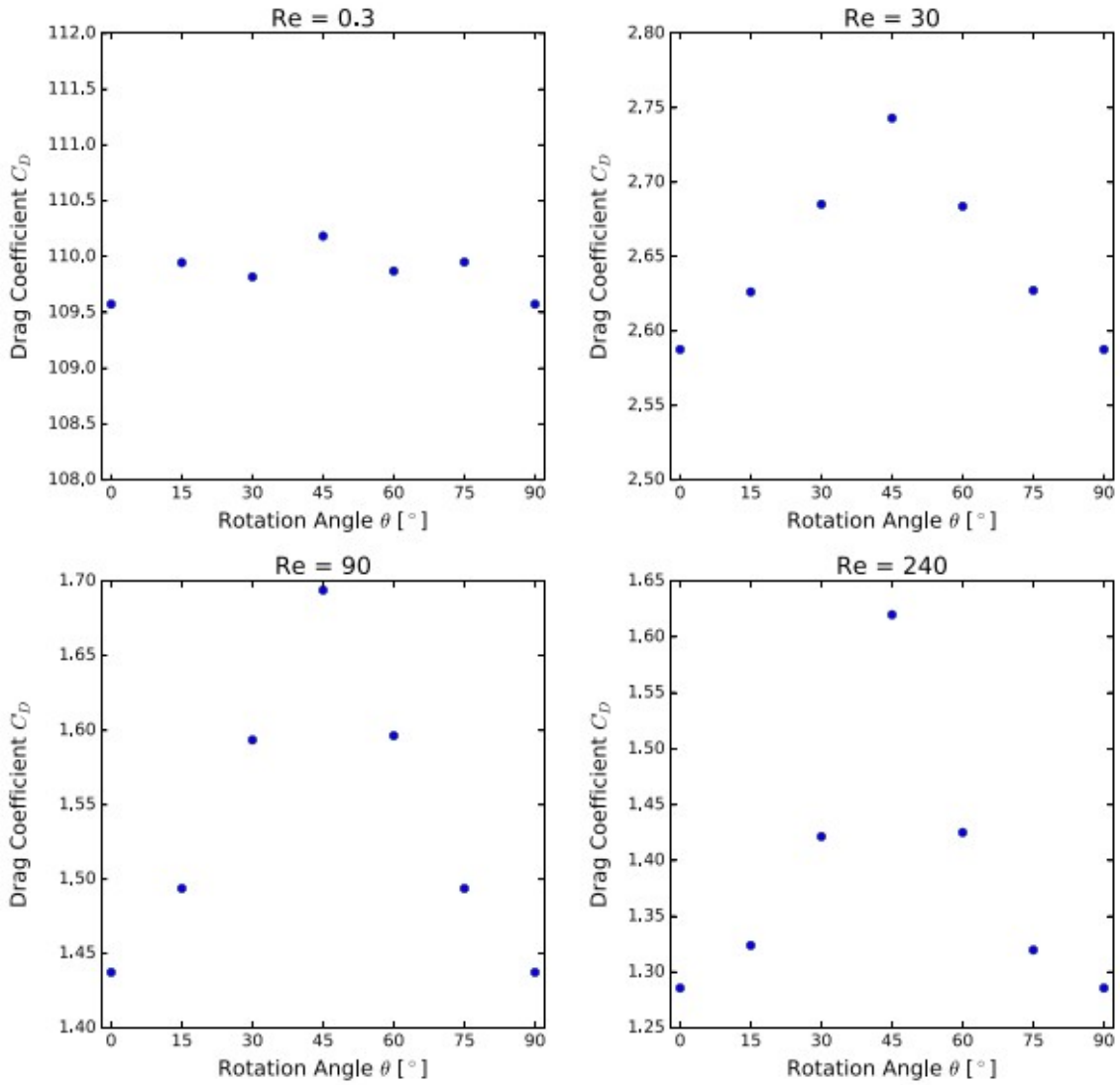


Figure 8. Drag coefficient C_D for cubes fixed in uniform flow as a function of rotation of cube relative to flow direction at Reynolds number = 0.3, 30, 90, and 240

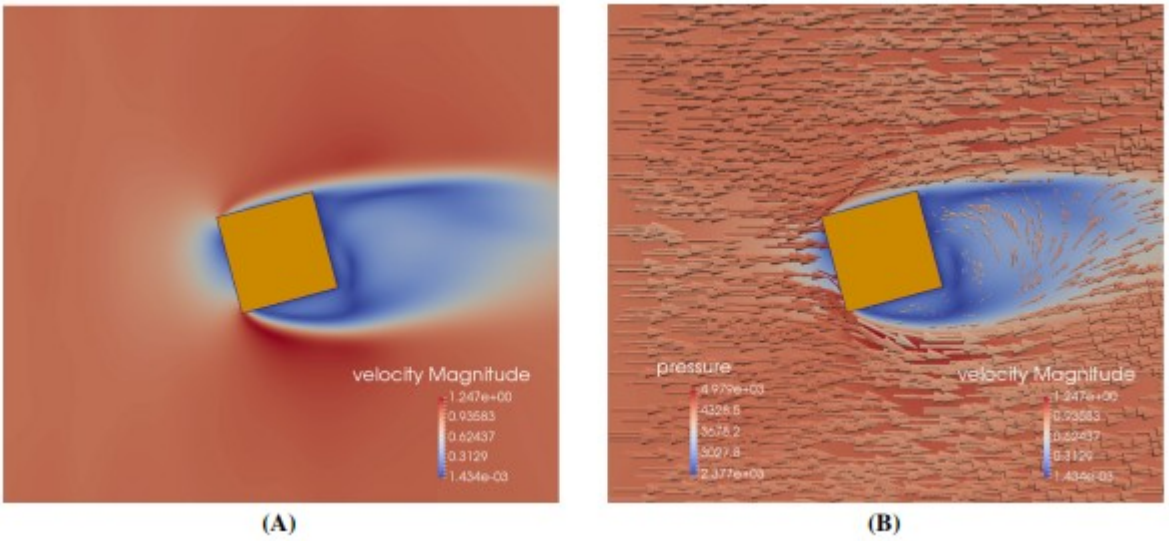


Figure 9. Cube rotated 15° relative to flow direction. The upstream boundary has constant velocity u=⟨1.0,0.0,0.0⟩ m/s while the downstream boundary is a non-reflecting characteristic boundary. All remaining boundaries are periodic. A, Velocity field magnitude with included legend given in units of m/s. B, Velocity field vectors scaled based on magnitude. Arrows are colored based on pressure where included legend is in *Pascals*

5.3 Uplift forces on hydraulic structures

Hydraulic uplift forces at the cracks and joints on slabs in hydraulic structures can pose a major risk to their safe and reliable operation. Offsets in the joints or cracks can cause the hydraulic pressures to be transmitted underneath the slabs causing uplift or erosion of the foundation materials.[62] The stagnation pressure and flow patterns associated with offset joints were simulated using the coupled DEM-LBM model to illustrate its potential use for this class of problems. For these simulations, a 1/8-inch joint with 1/8-inch offset between slabs was modeled, similar to the physical and numerical experiments performed by Frizell.[62] Figure 10 shows the boundaries for the simulations. The two slabs are separated in the center of the domain by the joint where the offset between the two slabs can be clearly seen. The left boundary is the upstream side where fluid enters the domain at a constant velocity and flows toward the right side of the domain where flow is forced upward by the offset joint.

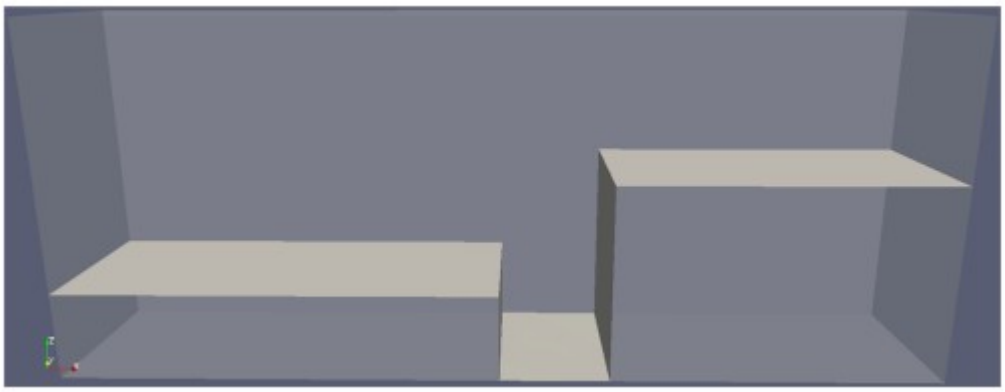
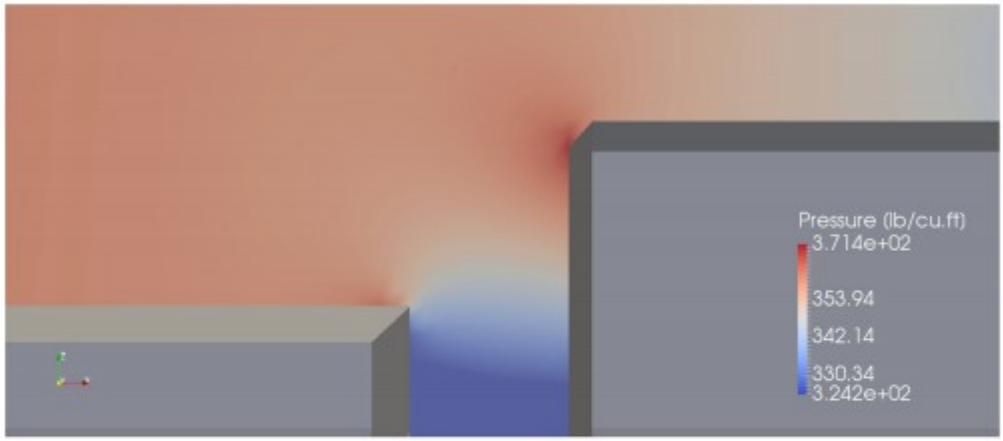
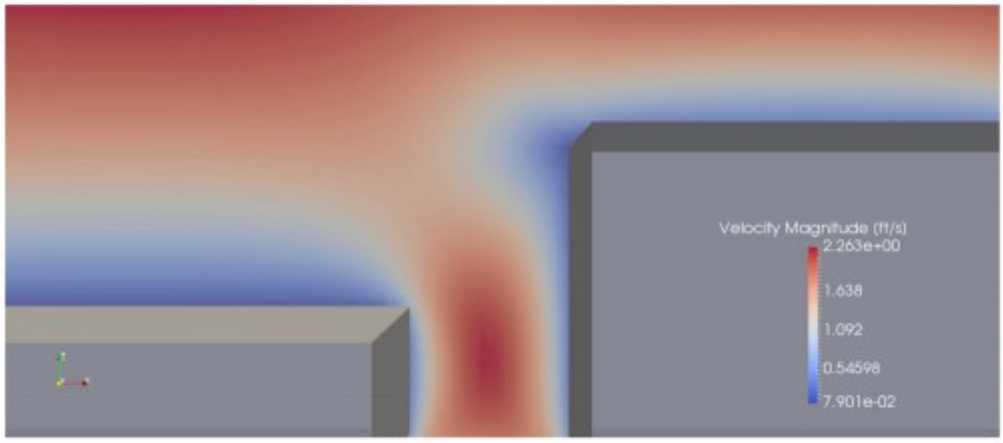


Figure 10. Two slabs offset by 1/8 inch separated by 1/8 inch joint

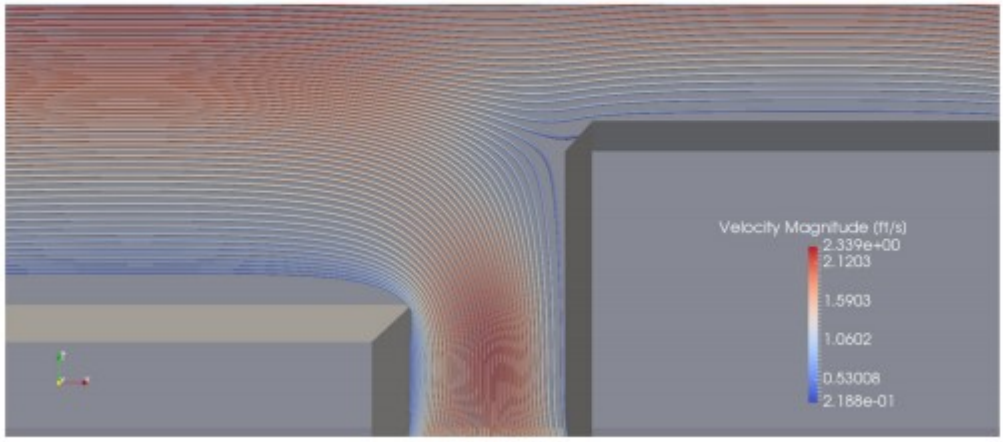
Two different cases were evaluated to show how the flow characteristics change depending on whether the joint is open—water can flow underneath the slabs—or sealed. The resulting pressure and velocity fields with associated stream tracers are shown for the open joint case in Figure 11 and for the sealed joint case in Figure 12. The location of the stagnation pressure and general flow characteristics agree well with the results presented in Frizell.[62]



(A)



(B)



(C)

Figure 11. Simulation results for 1/8-inch offset slabs with 1/8-inch open joint where water is able to flow through the joint. A, Pressure field in and around the joint, B, velocity field magnitude, and C, velocity field tracers

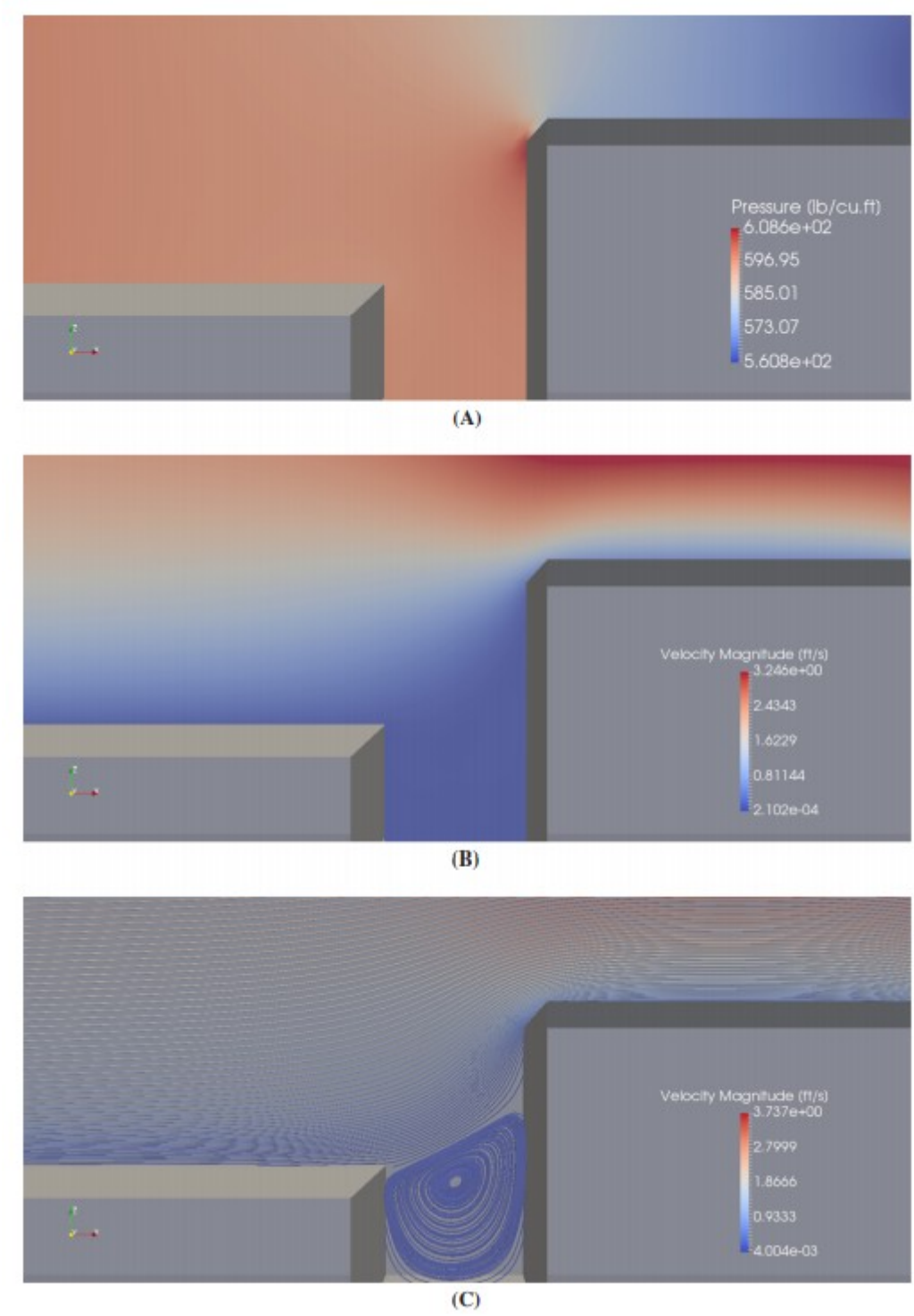


Figure 12. Simulation results for 1/8-inch offset slabs with 1/8-inch closed joint. A, Pressure field in and around the joint, B, velocity field magnitude, and C, velocity field tracers

5.4 Sedimentation of multiple polygonal particles

The ultimate objective is to model the interaction between multiple polyhedral particles and fluid. As shown in Figure 13, four polyhedral

particles initially packed tightly together are allowed to free fall through fluid under gravitational loading. Figure 14 shows the velocity profile in the fluid as the blocks sink and come to rest on the lower rigid boundary. As mentioned previously, accurately capturing the shape of the particles in the analyses is essential for predicting the correct kinematic response. If the fluid mesh is too coarse, the particle shape will not be represented with sufficient resolution in the fluid solution. This can lead to erroneous results because the fluid forces are based on a misrepresentation of the actual particle shape.

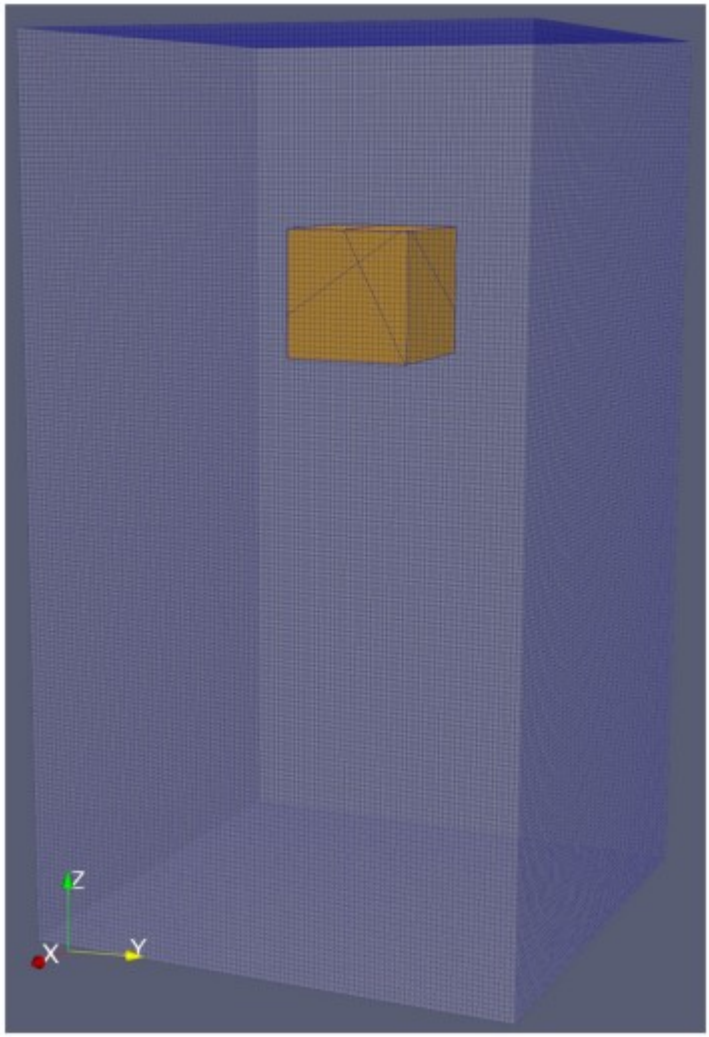


Figure 13. Four polyhedral blocks initially at rest inside fluid

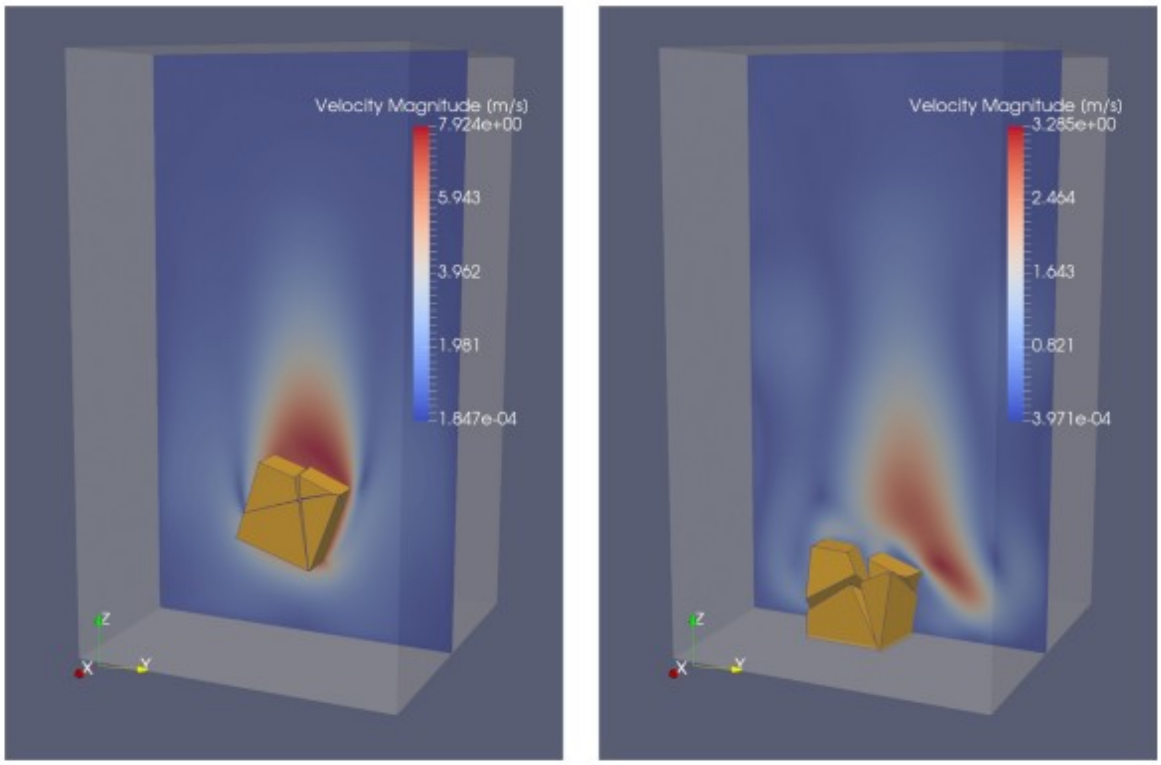


Figure 14. Sections through fluid as four polyhedral blocks sediment through it. A, Velocity profile around blocks as they accelerate because of gravity and B, blocks impacting lower boundary

While these analyses highlight the capability of the algorithm, they also illustrate a potential computational issue: the mesh spacing requirements are significantly different for densely packed particles compared with particles dispersed in fluid. Table 2 shows the mesh spacing, number of nodes, and computational times required for four different analyses. The computational bottleneck in these simulations is the fluid-solid coupling computations and it can be seen that the computation time significantly increases as the number of fluid nodes is increased. In order to accurately resolve the multiscale nature of the interactions when multiple particles are densely packed, the mesh size needs to be quite fine compared with when particles are further apart and not in contact. Here, global mesh refinement has been applied; however, the fine mesh size required in the vicinity of particle boundaries is not required in other parts of the domain and thus is not an efficient use of computational resources. To address this computational issue, adaptive meshing and multigrid methods are required which are currently being investigated.

TABLE 2 Sedimentation analyses computation configuration

$\Delta x[m]$	Number of LBM Nodes	Computation Time, Hours
0.2	15 876	8.5
0.1	119 351	10.3
0.05	925 101	44.5
0.04	1 795 376	109.6

6 CONCLUSIONS

A new coupling algorithm was developed to model the interaction between fluid and solid polyhedral particles where the particles are allowed to move through the fluid. The fluid phase was modeled using LBM while the solid particles were modeled using DEM. The coupling algorithm is based on a volume-fraction approach[9] where fluid-solid interaction is established through constrained optimization and simplex integration. The algorithm proceeds by first establishing whether the particle and fluid cell overlap—essentially a contact detection problem as encountered in DEM—through a linear programming approach.[35] In the case where the particle and fluid cell do overlap, the minimal set of constraints that describe the volume of the region of overlap is established. With this minimal set, the volumetric solid fraction is then calculated analytically using simplex integration.[58] The coupling algorithm was validated against experimental data.

The new coupling algorithm allows accurate evaluation of the fluid forces exerted on polyhedral particles. In particular, it can be used to consider sedimentation and particle transport where the polyhedral shape of particles influences the interaction between particles as well as the hydrodynamic loading. Additionally, the ability to capture the kinematic response of fractured rock based on particle shape, while also considering hydrodynamic loading, allows for analyses of many problems in geological engineering. The computational demands of these multiscale analyses where there is a large scale disparity between particle size and the pore space between the particles requires adaptive meshing and a parallel implementation of both DEM and LBM. As such, our current research is focusing on implementing the new algorithm in parallel with adaptive meshing in order to allow for analysis of more complex problems using high performance computing.

ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation (NSF) grant CMMI-1363354 and the Edward G. Cahill and John R. Cahill Endowed Chair funds.

APPENDIX A1: TRANSFORMATION MATRICES FOR MRT LBM

M used in the multi-relaxation-time collision operator described in Section 3 is calculated from Geier et al,[51] with the discrete velocities c_i listed

in Table 1, as shown in Figure 2, for the D3Q27 velocity set. The resulting tranformation matrix is calculated as:

$$\boldsymbol{M} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
-2 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 2 & -1 & -1 & 1 & 1 & 1 & 1 & -2 & -2 & 0 & 0 & 0 & 0 & 2 & -1 & -1 & 1 & 1 & 1 & 1 & -2 & -2 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & -4 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 2 & 2 & 2 & 2 & 4 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & -2 & -2 & -2 & -2 \\
0 & 0 & -4 & 0 & -1 & 1 & 0 & 0 & -1 & -1 & 2 & 2 & -2 & -2 & 0 & 4 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -2 & -2 & 2 & 2 \\
0 & 0 & 0 & -4 & 0 & 0 & -1 & 1 & -1 & 1 & 2 & -2 & 2 & -2 & 0 & 0 & 4 & 0 & 0 & 1 & -1 & 1 & -1 & -2 & 2 & -2 & 2 \\
0 & 4 & 0 & 0 & -2 & -2 & -2 & -2 & 0 & 0 & 1 & 1 & 1 & 1 & -4 & 0 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & -1 & -1 & -1 & -1 \\
0 & 0 & 4 & 0 & -2 & 2 & 0 & 0 & -2 & -2 & 1 & 1 & -1 & -1 & 0 & -4 & 0 & 2 & -2 & 0 & 0 & 2 & 2 & -1 & -1 & 1 & 1 \\
0 & 0 & 0 & 4 & 0 & 0 & -2 & 2 & -2 & 2 & 1 & -1 & 1 & -1 & 0 & 0 & -4 & 0 & 0 & 2 & -2 & 2 & -2 & -1 & 1 & -1 & 1 \\
4 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
-8 & 4 & 4 & 4 & -2 & -2 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 & 4 & 4 & 4 & -2 & -2 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\
0 & -4 & 2 & 2 & 1 & 1 & 1 & 1 & -2 & -2 & 0 & 0 & 0 & 0 & -4 & 2 & 2 & 1 & 1 & 1 & 1 & -2 & -2 & 0 & 0 & 0 & 0 \\
0 & 0 & -2 & 2 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1
\end{bmatrix} \tag{A1}$$

and its inverse is calculated as:

$$\boldsymbol{M}^{-1} = \frac{1}{216} \begin{bmatrix}
8 & 0 & 0 & 0 & -24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24 & -8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & 12 & 0 & 0 & -12 & 12 & 0 & 0 & 0 & 0 & -12 & 0 & 0 & 12 & 0 & 0 & 0 & 4 & -12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & 0 & 12 & 0 & -12 & -6 & 18 & 0 & 0 & 0 & 0 & -12 & 0 & 0 & 12 & 0 & 0 & 4 & 6 & -18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & 0 & 0 & 12 & -12 & -6 & -18 & 0 & 0 & 0 & 0 & 0 & -12 & 0 & 0 & 12 & 0 & 4 & 6 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & 12 & 12 & 0 & 0 & 6 & 18 & 18 & 0 & 0 & -3 & -3 & 0 & -6 & -6 & 0 & -6 & -2 & 3 & 9 & -18 & 0 & 0 & 27 & -27 & 0 & 0 \\
8 & 12 & -12 & 0 & 0 & 6 & 18 & -18 & 0 & 0 & -3 & 3 & 0 & -6 & 6 & 0 & -6 & -2 & 3 & 9 & 18 & 0 & 0 & 27 & 27 & 0 & 0 \\
8 & 12 & 0 & 12 & 0 & 6 & -18 & 0 & 0 & 18 & -3 & 0 & -3 & -6 & 0 & -6 & -6 & -2 & 3 & -9 & 0 & 0 & -18 & -27 & 0 & 27 & 0 \\
8 & 12 & 0 & -12 & 0 & 6 & -18 & 0 & 0 & -18 & -3 & 0 & 3 & -6 & 0 & 6 & -6 & -2 & 3 & -9 & 0 & 0 & 18 & -27 & 0 & -27 & 0 \\
8 & 0 & 12 & 12 & 0 & -12 & 0 & 0 & 18 & 0 & 0 & -3 & -3 & 0 & -6 & -6 & -6 & -2 & -6 & 0 & 0 & -18 & 0 & 0 & 27 & -27 & 0 \\
8 & 0 & 12 & -12 & 0 & -12 & 0 & 0 & -18 & 0 & 0 & -3 & 3 & 0 & -6 & 6 & -6 & -2 & -6 & 0 & 0 & 18 & 0 & 0 & 27 & 27 & 0 \\
8 & 12 & 12 & 12 & 12 & 0 & 0 & 18 & 18 & 18 & 6 & 6 & 6 & 3 & 3 & 3 & 6 & 1 & 0 & 0 & 9 & 9 & 9 & 0 & 0 & 0 & 27 \\
8 & 12 & 12 & -12 & 12 & 0 & 0 & 18 & -18 & -18 & 6 & 6 & -6 & 3 & 3 & -3 & 6 & 1 & 0 & 0 & 9 & -9 & -9 & 0 & 0 & 0 & -27 \\
8 & 12 & -12 & 12 & 12 & 0 & 0 & -18 & -18 & 18 & 6 & -6 & 6 & 3 & -3 & 3 & 6 & 1 & 0 & 0 & -9 & -9 & 9 & 0 & 0 & 0 & -27 \\
8 & 12 & -12 & -12 & 12 & 0 & 0 & -18 & 18 & -18 & 6 & -6 & -6 & 3 & -3 & -3 & 6 & 1 & 0 & 0 & -9 & 9 & -9 & 0 & 0 & 0 & 27 \\
8 & -12 & 0 & 0 & -12 & 12 & 0 & 0 & 0 & 0 & 12 & 0 & 0 & -12 & 0 & 0 & 0 & 4 & -12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & 0 & -12 & 0 & -12 & -6 & 18 & 0 & 0 & 0 & 0 & 12 & 0 & 0 & -12 & 0 & 0 & 4 & 6 & -18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & 0 & 0 & -12 & -12 & -6 & -18 & 0 & 0 & 0 & 0 & 0 & 12 & 0 & 0 & -12 & 0 & 4 & 6 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
8 & -12 & -12 & 0 & 0 & 6 & 18 & 18 & 0 & 0 & 3 & 3 & 0 & 6 & 6 & 0 & -6 & -2 & 3 & 9 & -18 & 0 & 0 & -27 & 27 & 0 & 0 \\
8 & -12 & 12 & 0 & 0 & 6 & 18 & -18 & 0 & 0 & 3 & -3 & 0 & 6 & -6 & 0 & -6 & -2 & 3 & 9 & 18 & 0 & 0 & -27 & -27 & 0 & 0 \\
8 & -12 & 0 & -12 & 0 & 6 & -18 & 0 & 0 & 18 & 3 & 0 & 3 & 6 & 0 & 6 & -6 & -2 & 3 & -9 & 0 & 0 & -18 & 27 & 0 & -27 & 0 \\
8 & -12 & 0 & 12 & 0 & 6 & -18 & 0 & 0 & -18 & 3 & 0 & -3 & 6 & 0 & -6 & -6 & -2 & 3 & -9 & 0 & 0 & 18 & 27 & 0 & 27 & 0 \\
8 & 0 & -12 & -12 & 0 & -12 & 0 & 0 & 18 & 0 & 0 & 3 & 3 & 0 & 6 & 6 & -6 & -2 & -6 & 0 & 0 & -18 & 0 & 0 & -27 & 27 & 0 \\
8 & 0 & -12 & 12 & 0 & -12 & 0 & 0 & -18 & 0 & 0 & 3 & -3 & 0 & 6 & -6 & -6 & -2 & -6 & 0 & 0 & 18 & 0 & 0 & -27 & -27 & 0 \\
8 & -12 & -12 & -12 & 12 & 0 & 0 & 18 & 18 & 18 & -6 & -6 & -6 & -3 & -3 & -3 & 6 & 1 & 0 & 0 & 9 & 9 & 9 & 0 & 0 & 0 & -27 \\
8 & -12 & -12 & 12 & 12 & 0 & 0 & 18 & -18 & -18 & -6 & -6 & 6 & -3 & -3 & 3 & 6 & 1 & 0 & 0 & 9 & -9 & -9 & 0 & 0 & 0 & 27 \\
8 & -12 & 12 & -12 & 12 & 0 & 0 & -18 & 18 & 18 & -6 & 6 & -6 & -3 & 3 & -3 & 6 & 1 & 0 & 0 & -9 & 9 & 9 & 0 & 0 & 0 & 27 \\
8 & -12 & 12 & 12 & 12 & 0 & 0 & -18 & 18 & -18 & -6 & 6 & 6 & -3 & 3 & 3 & 6 & 1 & 0 & 0 & -9 & 9 & -9 & 0 & 0 & 0 & -27
\end{bmatrix} \tag{A2}$$

REFERENCES

1 Anderson TB, Jackson R. Fluid mechanical description of fluidized beds. Equations of motion. *Ind Eng Chem Fundam*. 1967; 6(4): 527- 539.

2 Tsuji Y, Kawaguchi T, Tanaka T. Discrete particle simulation of two-dimensional fluidized bed. *Powder Technol*. 1993; 77(1): 79- 87.

3 Xu BH, Yu A. Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics. *Chem Eng Sci*. 1997; 52(16): 2785- 2809.

4 Tsuji T, Ito A, Tanaka T. Multi-scale structure of clustering particles. *Powder Technol*. 2008; 179(3): 115- 125. WCPT5.

5 Mikito F, Daisuke N. Robust coupled fluid-particle simulation scheme in stokes-flow regime: Toward the geodynamic simulation including granular media. *Geochem Geophys Geosyst*. 2014; 15(7): 2865- 2882.

6 Guo N, Zhao J. Parallel hierarchical multiscale modelling of hydro-mechanical problems for saturated granular soils. *Comput Meth Appl Mech Eng*. 2016; 305: 37- 61.

7 Wang K, Sun W. A semi-implicit discrete-continuum coupling method for porous media based on the effective stress principle at finite strain. *Comput Meth Appl Mech Eng*. 2016; 304: 546- 583.

8 Wang K, Sun W. An updated Lagrangian LBM-DEM-FEM coupling model for dual-permeability fissured porous media with embedded discontinuities. *Comput Meth Appl Mech Eng*. 2019; 344: 276- 305.

9 Noble DR, Torczynski JR. A lattice-boltzmann method for partially saturated computational cells. *Int J Mod Phys C*. 1998; 09(08): 1189- 1201.

10 Holdych DJ. Lattice Boltzmann methods for diffuse and mobile interfaces. *PhD Thesis*. University of Illinois at Urbana-Champaign; 2003.

11 Owen DRJ, Leonardi CR, Feng YT. An efficient framework for fluid-structure interaction using the lattice Boltzmann method and immersed moving boundaries. *International Journal for Numerical Methods in Engineering*. 2010; 87(1-5): 66- 95.

12 Erik SO, K. CB. Three-dimensional immersed boundary conditions for moving solids in the lattice-boltzmann method. *Int J Numer Methods Fluids*. 2007; 55(2): 103- 125.

13 Galindo–Torres SA. A coupled discrete element lattice Boltzmann method for the simulation of fluid-solid interaction with particles of general shapes. *Comput Meth Appl Mech Eng*. 2013; 265: 107- 119.

14 Cundall PA, Strack ODL. A discrete numerical model for granular assemblies. *Gèotechnique*. 1979; 29(1): 47- 65.

15 Cundall PA. Formulation of a three-dimensional distinct element model—Part I. A scheme to detect and represent contacts in a system composed of many polyhedral blocks. *Int J Rock Mech Min Sci Geomech Abstr*. 1988; 25(3): 107- 116.

16 Hart R, Cundall PA, Lemos J. Formulation of a three-dimensional distinct element model—Part II. Mechanical calculations for motion and interaction of a system composed of many polyhedral blocks. *Int J Rock Mech Min Sci Geomech Abstr*. 1988; 25(3): 117- 125.

17 Shi GH. Discontinuous deformation analysis—A new model for the statics and dynamics of block systems. *PhD Thesis*. University of California, Berkeley; 1988.

18 Shi GH, Goodman RE. Discontinuous deformation analysis—A new method for computing stress, strain and sliding of block systems. In: The 29th US Symposium on Rock Mechanics (USRMS) American Rock Mechanics Association; 1988.

19 Zienkiewicz OC, Taylor RL, Nithiarasu P. *The Finite Element Method for Fluid Dynamics*: Oxford : Butterworth-Heinemann, 2014; 2014.

20 LeVeque RJ. *Finite Volume Methods for Hyperbolic Problems*: Cambridge University Press; 2002.

21 McNamara GR, Zanetti G. Use of the Boltzmann equation to simulate lattice-gas automata. *Phys Rev Lett*. 1988; 61(20): 2332.

22 Succi S, Foti E, Higuera F. Three-dimensional flows in complex geometries with the lattice boltzmann method. *EPL (Europhys Lett)*. 1989; 10(5): 433.

23 Bouzidi M, Firdaouss M, Lallemand P. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Phys Fluids*. 2001; 13(11): 3452- 3459.

24 Boutt DF, Cook BK, McPherson BJOL, Williams JR. Direct simulation of fluid-solid mechanics in porous media using the discrete element and lattice-Boltzmann methods. *J Geophys Res Solid Earth*. 2007; 112(B10).

25 Han Y, Cundall PA. Lbm-DEM modeling of fluid-solid interaction in porous media. *Int J Numer Anal Methods Geomech*. 2013; 37(10): 1391- 1407.

26 Lominé F, Scholtès L, Sibille L, Poullain P. Modeling of fluid-solid interaction in granular media with coupled lattice Boltzmann/discrete element methods: application to piping erosion. *Int J Numer Anal Methods Geomech*. 2013; 37(6): 577- 596.

27 Johnson DH, Vahedifard F, Jelinek B, Peters JF. Micromechanics of undrained response of dilative granular media using a coupled dem-LBM model: A case of biaxial test. *Comput Geol*. 2017; 89: 103- 112.

28 Guan Y, Guadarrama-Lara R, Jia X, Zhang K, Wen D. Lattice boltzmann simulation of flow past a non-spherical particle. *Adv Powder Technol*. 2017; 28(6): 1486- 1494.

29 Sommerfeld M, Qadir Z. Fluid dynamic forces acting on irregular shaped particles: Simulations by the lattice-Boltzmann method. *Int J Multiphase Flow*. 2018; 101: 212- 222.

30 Wang M, Feng YT, Pande GN, Zhao TT. A coupled 3-dimensional bonded discrete element and lattice Boltzmann method for fluid-solid coupling in cohesive geomaterials. *Int J Numer Anal Methods Geomech*. 2018; 42(12): 1405- 1424.

31 Krüger T, Kusumaatmaja H, Kuzmin A, Shardt O, Silva G, Viggen EM. *The Lattice Boltzmann Method*: Springer; 2017.

32 d'Humières D, Ginzburg I, Krafczyk M, Lallemand P, Luo LS. Multiple–relaxation–time lattice Boltzmann models in three dimensions. *Philos Trans R Soc A Math Phys Eng Sci*. 2002; 360(1792): 437- 451.

33 Horner DA, Peters JF, Carrillo A. Large scale discrete element modeling of vehicle-soil interaction. *J Eng Mech*. 2001; 127(10): 1027- 32.

34 Williams JR, Perkins E, Cook B. A contact algorithm for partitioning N arbitrary sized objects. *Eng Comput*. 2004; 21(2/3/4): 235- 48.

35 Boon CW, Houlsby GT, Utili S. A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method. *Comput Geol*. 2012; 44: 73- 2.

36 Walton OR, Braun RL. Viscosity, granular-temperature, and stress calculations for shearing assemblies of inelastic, frictional disks. *J Rheol*. 1986; 30(5): 949- 80.

37 Thornton C, Yin KK. Impact of elastic spheres with and without adhesion. *Powder Technol*. 1991; 65(1): 153- 66. A Special Volume Devoted to the Second Symposium on Advances in Particulate Technology.

38 Vu-Quoc L, Zhang X. An accurate and efficient tangential force-displacement model for elastic frictional contact in particle-flow simulations. *Mech Mater*. 1999; 31(4): 235- 9.

39 Goodman RE, Kieffer DS. Behavior of rock in slopes. *J Geotech Geoenviron Eng*. 2000; 126(8): 675- 84.

40 Barton N, Bandis S. Rock joint model for analyses of geologic discontinua. In: Proc. 2rid Int. Conf. on Constitutive Laws for Engng. Materials; 1987.

41 Plesha ME. Constitutive models for rock discontinuities with dilatancy and surface degradation. *Int J Numer Anal Methods Geomech*. 1987; 11(4): 345- 62.

42 Amadei B, Saeb S. Constitutive Models of Rock Joints. In: Rock Joints. Proceedings of the International Symposium on Rock Joints, Loen, Norway; 1990: 4- 6.

43 Jing L, Nordlund E, Stephansson O. A 3-D constitutive model for rock joints with anisotropic friction and stress dependency in shear stiffness. *Int J Rock Mech Min Sci Geomech Abstr*. 1994; 31(2): 173- 178.

44 Swope WC, Andersen HC, Berens PH, Wilson KR. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: application to small water clusters. *J Chem Phys*. 1982; 76(1): 637- 649.

45 Johnson SM, Williams JR, Cook BK. Quaternion - based rigid body rotation integration algorithms for use in particle methods. *Int J Numer Methods Eng*. 2007; 74(8): 1303- 1313.

46 Suga K, Kuwata Y, Takashima K, Chikasue R. A D3Q27 multiple-relaxation-time lattice Boltzmann method for turbulent flows. *Comput Math Appl*. 2015; 69(6): 518- 529.

47 Kang SK, Hassan YA. The effect of lattice models within the lattice Boltzmann method in the simulation of wall-bounded turbulent flows. *J Comput Phys*. 2013; 232(1): 100- 117.

48 Kuwata Y, Suga K. Anomaly of the lattice Boltzmann methods in three-dimensional cylindrical flows. *J Comput Phys*. 2015; 280: 563- 569.

49 Bhatnagar PL, Gross EP, Krook M. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys Rev*. 1954; 94: 511- 525.

50 Luo LS. Unified theory of lattice Boltzmann models for nonideal gases. *Phys Rev Lett*. 1998; 81: 1618- 1621.

51 Geier M, Schönherr M, Pasquali A, Krafczyk M. The cumulant lattice Boltzmann equation in three dimensions: Theory and validation. *Comput Math Appl*. 2015; 70(4): 507- 547.

52 Guo Z, Zheng C, Shi B. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys Rev E*. 2002; 65(4): 046308.

53 Li H, Pan C, Miller CT. Pore-scale investigation of viscous coupling effects for two-phase flow in porous media. *Phys Rev E*. 2005; 72: 026705.

54 Zou Q, He X. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Phys Fluids*. 1997; 9(6): 1591- 1598.

55 Boon CW, Houlsby GT, Utili S. A new rock slicing method based on linear programming. *Comput Geotech*. 2015; 65: 12- 29.

56 Brearley AL, Mitra G, Williams HP. Analysis of mathematical programming problems prior to applying the simplex algorithm. *Math Program*. 1975; 8(1): 54- 83.

57 Caron RJ, McDonald JF, Ponic CM. A degenerate extreme point strategy for the classification of linear constraints as redundant or necessary. *J Optim Theory Appl*. 1989; 62(2): 225- 237.

58 Shi GH. Working Forum on Manifold Method of Material Analysis, Volume 2. The Numerical Manifold Method and Simplex Integration, DTIC Document; 1997.

59 Haider A, Levenspiel O. Drag coefficient and terminal velocity of spherical and nonspherical particles. *Powder Technol*. 1989; 58(1): 63- 70.

60 Ganser GH. A rational approach to drag prediction of spherical and nonspherical particles. *Powder Technol*. 1993; 77(2): 143- 152.

61 Hölzer A, Sommerfeld M. New simple correlation formula for the drag coefficient of non-spherical particles. *Powder Technol*. 2008; 184(3): 361- 365.

62 Frizell W. Uplift and crack flow resulting from high velocity discharges over open offset joints—laboratory studies. DSO-07-07, United States Bureau of Reclamation; 2007.