Modeling of Dynamic Rock–Fluid Interaction Using Coupled 3-D Discrete Element and Lattice Boltzmann Methods

Michael Gardner[1] · Nicholas Sitar[2]

[1] NHERI SimCenter, University of California, Berkeley, CA 94720, USA [2] Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720, USA

Michael Gardner: mhgardner@berkeley.edu

Abstract

Scour of rock is a challenging and interesting problem that combines rock mechanics and hydraulics of turbulent flow. On a practical level, rock erosion is a critical issue facing many of the world's dams at which excessive scour of the dam foundation or spillway can compromise the stability of the dam resulting in significant remediation costs, if not direct personal property damage or even loss of life. This interaction between the blocky rock mass and water is analyzed by directly modeling the solid and fluid phases—the individual polyhedral blocks are modeled using the discrete element method (DEM) while water is modeled using the lattice Boltzmann method (LBM). The LBM mesh is entirely independent of the DEM discretization, making it possible to refine the LBM mesh such that transient and varied fluid pressures acting on the rock surfaces are directly modeled. This provides the capability to investigate the effect of water pressure inside the fractured rock mass, along potential sliding planes, and can be extended to rock falls and slides into standing bodies of water such as lakes and reservoirs. Results show that the coupled DEM–LBM implementation is able to accurately capture the interaction between polyhedral rock blocks and fluid by analytically solving for the solid volume fraction in the coupling computations using convex optimization and simplex integration; however, further performance improvements are necessary to simulate realistic, field-scale problems. Particularly, adaptive mesh refinement and multigrid methods implemented in a parallel computing environment will be essential for capturing the highly computationally intensive and multiscale nature of rock–fluid interaction.

Keywords Rock scour · Fluid–solid interaction · Discrete element method · Lattice Boltzmann method · Linear programming · Simplex integration

1 Introduction

Erosion of rock by hydraulic forces—scour and plucking—is an important element of natural evolution of the landscape and this process can have significant impact on engineered structures. Unlined dam spillways, bridge abutments, and tunnels are all subject to water erosion and the resulting damage can cost millions of dollars to repair and, in the worst case, cause loss of life.

A recent example of the impact of rock scour and the speed with which it can occur is the Oroville Dam, located in the foothills of the Sierra Nevada in Northern California. In early February, 2017, the concrete chute of the flood control spillway failed and the underlying rock experienced rapid erosion as water was discharged at high velocity and volumes up to 100,000 cubic feet per second. Figure 1 shows the extent of the damage caused to the spillway and the erosion of the underlying rock mass by hydraulic scour and plucking in roughly 2 weeks.

Fig. 1 Scour damage to service spillway at Oroville Dam in Northern California, 26 February 2017 (Kolke 2017)

This incident has made dam owners acutely aware of the risk posed by the potential for rock scour and the need to re-evaluate the safety of these structures. This requires an understanding of the mechanisms involved and the ability to fully analyze the hydraulics of the problem. As outlined by George and Sitar (2016), currently used empirical methods of scour analysis tend to over simplify the three-dimensional nature of fractured rock masses and may miss the influence of geologic structure on rock mass erodibility.

In our effort to develop a simulation tool capable of capturing the three-dimensional nature of the problem, we chose to adopt the discrete element method (DEM) (Cundall and Strack 1979; Cundall 1988; Hart et al. 1988). The explicit, decoupled nature of DEM makes it attractive because the scale of the numerical problem necessitates parallel computations. Only information on blocks' immediate neighbors is required to calculate contact forces after which the motion of each block can be updated independently. Additionally, the three-dimensional formulation for polyhedral blocks is relatively straightforward.

Capturing the interaction between the blocky rock mass and the water flowing over and through it requires coupling between the numerical models for the solid and the fluid phases. This means that the hydrodynamic forces and moments exerted on the blocks need to be accounted for when

integrating the equations of motion for the solid phase while the effect of the particles in and moving through the fluid also needs to be incorporated into the fluid solver. Simulations that capture this interaction generally follow two approaches. The first approach incorporates fluid–solid interaction based on a locally averaged interaction between the two phases (Anderson and Jackson 1967; Tsuji et al. 1993; Xu and Yu 1997; Tsuji et al. 2008; Mikito and Daisuke 2014) while the second approach directly simulates hydrodynamic forces on the solid particles (Noble and Torczynski 1998; Holdych 2003; Owen et al. 2010; Strack and Cook 2007). In the locally averaged approach, the fluid–solid coupling is performed 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 to direct simulation since the number of solid particles is greater than the number of fluid cells. This approach may be appropriate in applications where only the averaged fluid–solid interaction is of interest, but it does not offer sufficient resolution when trying to establish the hydrodynamic interaction for individual blocks. Direct simulation of the fluid–solid interaction attempts to overcome this shortcoming by having a much higher density fluid mesh compared to 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.

Since the individual behavior of rock blocks is important for the kinematic response of fractured rock, the direct simulation of rock–fluid interaction is necessary. Methods conventionally used in computational fluid dynamics (CFD), such as the finite element method (FEM) (Zienkiewicz et al. 2014) and the finite volume method (FVM) (LeVeque 2002), are able to directly solve the hydrodynamic forces and moments acting on solid particles. However, when blocks are allowed to move through the mesh, these methods can be computationally prohibitively expensive due to the need to remesh based on updated positions of the blocks. In comparison, the lattice Boltzmann method (LBM) (McNamara and Zanetti 1988; Succi et al. 1989) allows for fluid–solid coupling quite efficiently. Additionally, LBM is localized in its formulation, making it amenable to adaptive remeshing and parallel computing—a necessity for direct simulation of fluid–solid interaction. Consequently, LBM was selected to model rock–water interaction. In this paper, we give an overview of the modeling methodology while concentrating on the implementation and example applications. Sections 2 and 3 provide background on DEM and LBM, respectively. Section 4 provides details on the fluid–solid coupling algorithm, which synthesizes MRT LBM, constrained optimization and simplex integration. Section 5 provides validation of our implementation and presents example applications.

2 Solid Phase Model

Given the inherent blocky nature of jointed and fractured rock masses, DEM (Cundall 1988; Hart et al. 1988) provides a convenient approach for explicitly modeling the polyhedral shape of the individual blocks. The DEM formulation

considers the motion of each block individually, incorporating the interaction of the block with its neighbors through contact forces. Once the contact forces between blocks are established, the motion of each block can be updated independently.

2.1 Contact Detection

Contact detection consists of two separate steps: neighbor search and contact resolution. During the neighbor search, the blocks that are close enough to possibly be in contact within a given time period or step are identified. Each of the nearest neighbors is then checked to resolve if the blocks are physically in contact. We have implemented the CGRID algorithm (Williams et al. 2004) to perform the neighbor search. CGRID falls within the spatial binning class of neighbor search algorithms which generally are $O(N)$. CGRID is able to maintain performance even when the sizes of particles in the simulation differ significantly, as is often the case for fractured rock, and the method is readily adaptable to three dimensions.

Next, the contact resolution phase determines which blocks do physically contact. To avoid unnecessary computations, the bounding spheres of the neighboring blocks are first checked for overlap. If they do not overlap, the blocks are not in contact. If they do, further computations are required. We have used a contact detection algorithm based on linear programming (Boon et al. 2012). This algorithm simplifies the contact detection process through use of the normal to each of the block faces and the distance of each face from a local origin.

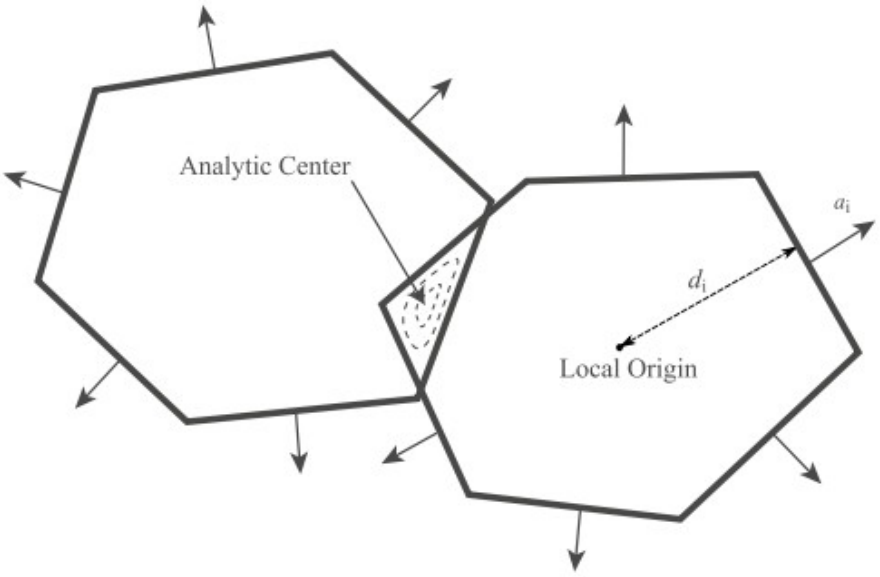


Fig. 2 Two colliding blocks with analytic center taken as contact point. Arrows indicate the direction of normal vectors to particle faces. The normal vector, a_i, for a particle face and the distance, d_i, of that face from a local origin are shown for one of the particle faces (modified, based on Boon et al. 2012)

The polyhedral block shape, shown in Fig. 2, is defined by N bounding planes such that

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

where $\boldsymbol{a_i}$ represents the normal vector to the ith plane bounding the block and $\boldsymbol{d_i}$ is the distance of that plane from some local origin—the origin can be at any location as long as all distances, $\boldsymbol{d_i}$, are referenced to the same location. Contact between two blocks is then established by solving the following linear program:

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

where $\boldsymbol{N_A}$ and $\boldsymbol{N_B}$ are the number of planes of the two neighboring blocks. The two blocks are in contact if $s<-\varepsilon$, where ε is a specified numerical tolerance. If the blocks are in contact, the contact point is taken as the analytic center of the region of overlap between the two contacting blocks. The analytic center, shown in Fig. 2, is calculated using the log-barrier method with Newton's method. The contact point, along with "potential particles" (Houlsby 2009) located entirely inside the blocks, are used to calculate the contact normal and overlap.

2.2 Contact Forces and Moments

The contact normal and overlap calculated in the contact resolution phase are used to establish the interaction forces between the blocks. In this research, the force and moments acting on the blocks were calculated following the approach outlined by Hart et al. (1988). Once the forces and moments due to inter-particle contacts have been calculated, the block positions are updated with the equations of motion for an individual block given by

$$\begin{aligned} \ddot{\boldsymbol{x}}_i + \alpha \dot{\boldsymbol{x}}_i &= \frac{\boldsymbol{F}_i}{m_i} + \boldsymbol{g}_i \\ \dot{\boldsymbol{\omega}}_i + \alpha \boldsymbol{\omega}_i &= \frac{\boldsymbol{M}_i}{I_i}, \end{aligned} \tag{3}$$

where $\ddot{\boldsymbol{x}}_i$ and $\dot{\boldsymbol{\omega}}_i$ are the translational and rotational acceleration of block i; $\boldsymbol{F_i}$ and $\boldsymbol{M_i}$ are the total force and moment acting on block i; α is a damping constant; m_i and I_i are the mass and moment of inertia of block i; and $\boldsymbol{g_i}$ is the gravitational acceleration block i. The block translation is integrated using a velocity Verlet finite difference approach (Swope et al. 1982), which is used in most DEM implementations, while the rotational motion is updated using a quaternion-based fourth-order Runge–Kutta approach (Johnson et al. 2007). The quaternion-based rotation integrator avoids the need to re-orthogonalize the rotation matrix, while also offering decreased memory usage and number of floating point operations.

3 Fluid Phase Model

The LBM arrives at the solution of the Navier–Stokes equations by describing the mesoscopic behavior of the fluid by considering the behavior of

distributions of particles rather than the macroscopic velocity and density or the behavior of individual particles. The physical basis of LBM is rooted in the Boltzmann equation, but it can be linked to the macroscopic behavior of fluid (Succi 2001), in effect solving the weakly compressible Navier–Stokes equations. LBM is attractive for modeling fluid–solid interaction because of the localized nature of the method and the ease with which complex shapes moving through the fluid domain can be accommodated. As a solid particle moves through the fluid domain, the status of the nodes that the solid interacts with is updated. Based on 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. Depending on the type of problem, there may not be a need for remeshing and the change in the status of each node is incorporated in the computations locally. In its formulation, LBM solves a discrete form of the Boltzmann equation—the so-called lattice Boltzmann equation (LBE) (McNamara and Zanetti 1988)—to arrive at the solution of various problems in fluid dynamics:

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

where $\boldsymbol{c_i}$ is the discrete set of velocities which limits the continuous particle velocity to a carefully selected subset. In this research, the D3Q27 velocity set (Suga et al. 2015), as shown in Fig. 3, was implemented. Equation 4 describes the streaming of particle populations $\boldsymbol{f_i}(\boldsymbol{x},\boldsymbol{t})$ moving with velocity $\boldsymbol{c_i}$ to a neighboring point located at $\boldsymbol{x}+\boldsymbol{c_i t}$ and the redistribution of particles through the collision operator $i\Omega_i$ at each point—this redistribution of particles models particle collisions. Streaming and collision constitute the fundamental concepts of the LBE, essentially advection and diffusion.

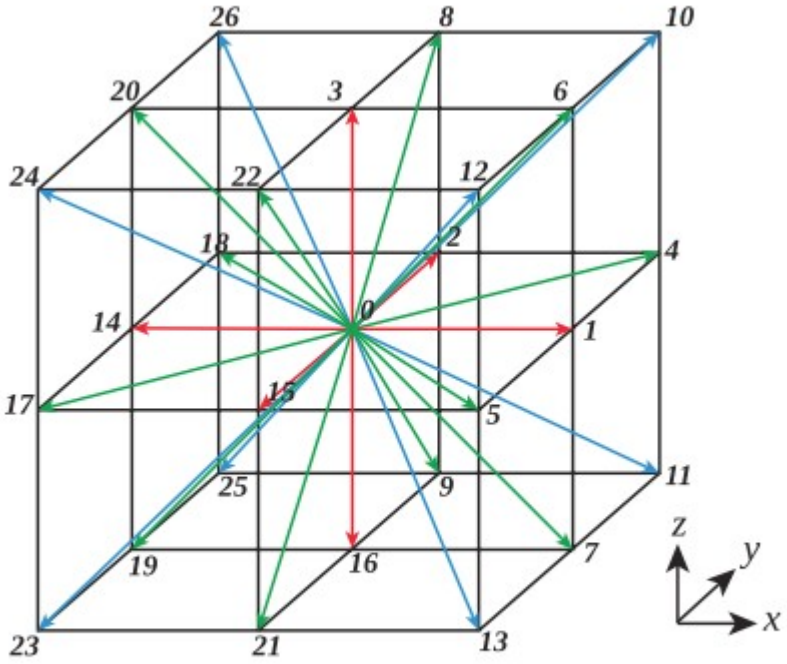


Fig. 3 D3Q27 velocity set showing discrete velocity ordering

The macroscopic fluid mass density and momentum are then calculated through weighted sums in velocity space, known as moments, of $\boldsymbol{f_i}$:

$$\begin{aligned} \rho(\boldsymbol{x},t) &= \sum_i f(\boldsymbol{x},t) \\ \rho\boldsymbol{u}(\boldsymbol{x},t) &= \sum_i \boldsymbol{c}_i f(\boldsymbol{x},t). \end{aligned} \tag{5}$$

The most commonly used collision operator is the Bhatnagar–Gross–Krook (BGK) collision operator (Bhatnagar et al. 1954):

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

This equation states that all populations $\boldsymbol{f_i}$ 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 (Luo 1998)

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

where $\boldsymbol{u}$ 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, $c_s = \Delta x/(\sqrt{3}\Delta t)$. The BGK collision operator relaxes all moments at the same rate, which leads to reduced numerical accuracy, particularly at large viscosities, and reduced stability, particularly at small viscosities (Krüger et al. 2017). However, the moments can all be relaxed at different rates using the multiple-relaxation-time (MRT) collision operator (d'Humières et al. 2002). Relaxing the moments at different rates achieves better stability and accuracy. The MRT form of the LBE is given by (d'Humières et al. 2002)

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

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

The transformation matrix $\boldsymbol{M}$ for the chosen velocity ordering was calculated using the orthogonal moment set from Geier et al. (2015) and 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. (2015). The macroscopic Navier–Stokes behavior is recovered when the kinematic shear viscosity ν is related to the corresponding components of $\hat{\boldsymbol{S}}$:

$$\nu = c_s^2\left(\frac{1}{s_5} - \frac{1}{2}\right)\Delta t = c_s^2\left(\frac{1}{s_7} - \frac{1}{2}\right)\Delta t. \tag{9}$$

3.1 Body Forces

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

$$f_i(\boldsymbol{x}+\boldsymbol{c}_i t, t+\Delta t) - f_i(\boldsymbol{x},t) = \Omega_i(\boldsymbol{x},t) + S_i(\boldsymbol{x},t). \tag{10}$$

The forcing scheme proposed by Guo et al. (2002) was implemented in this research where the equilibrium and macroscopic fluid velocity are defined as

$$\boldsymbol{u}^{\mathrm{eq}} = \boldsymbol{u} = \frac{1}{\rho}\sum_i f_i \boldsymbol{c}_i + \frac{\boldsymbol{F}\Delta t}{2\rho}, \tag{11}$$

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

$$S_i = \left(1 - \frac{\Delta t}{2\tau}\right) w_i \left(\frac{\boldsymbol{c}_i - \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u})\,\boldsymbol{c}_i}{c_s^4}\right) \cdot \boldsymbol{F}, \tag{12}$$

where $\boldsymbol{u}$ is the fluid velocity and τ is the BGK relaxation time. The approach proposed by Li et al. (2005) was used to include the relaxation parameters based on an MRT collision operator. The macroscopic fluid velocity is calculated by

$$\boldsymbol{u} = \frac{1}{\rho}\sum_i f_i \boldsymbol{c}_i + \frac{\boldsymbol{F}\Delta t}{2\rho}. \tag{13}$$

3.2 Turbulent Flow

Modeling turbulent flow requires special treatment because the grid scale may not be sufficiently small to capture all scales of the different flow features. The large eddy simulation (LES) approach is commonly used to account for this difference in the scale of flow structures—eddies greater than the grid scale are solved for directly while subgrid-scale flow structures are accounted for through a subgrid-scale (SGS) eddy viscosity, ν_{SGS}.

The wall-adapting local eddy viscosity (WALE) model (Nicoud and Ducros 1999) was implemented in our approach to calculate ν_{SGS}. In this model, the subgrid-scale viscosity is a function of the velocity gradient tensor, which can be calculated using second-order finite differences—this requires only knowing the fluid velocity of the nearest neighboring nodes. A full description of the method is provided in Nicoud and Ducros (1999). For the D3Q27 MRT model used, the subgrid-scale eddy viscosity calculated by the WALE model is added to the kinematic viscosity as follows (Suga et al. 2015):

$$\nu + \nu_{SGS} = c_s^2\left(\frac{1}{s_5} - \frac{1}{2}\right)\Delta t = c_s^2\left(\frac{1}{s_7} - \frac{1}{2}\right)\Delta t. \tag{14}$$

4 Fluid–Solid Coupling

The interaction between the fluid and solid phases requires coupling of two models, DEM and LBM, through exchange of information. The coupling process is achieved through a boundary condition based on the volume fraction of solid present in fluid cells. This approach allows polyhedral blocks to move through the fluid mesh while maintaining a similar form of the LBE.

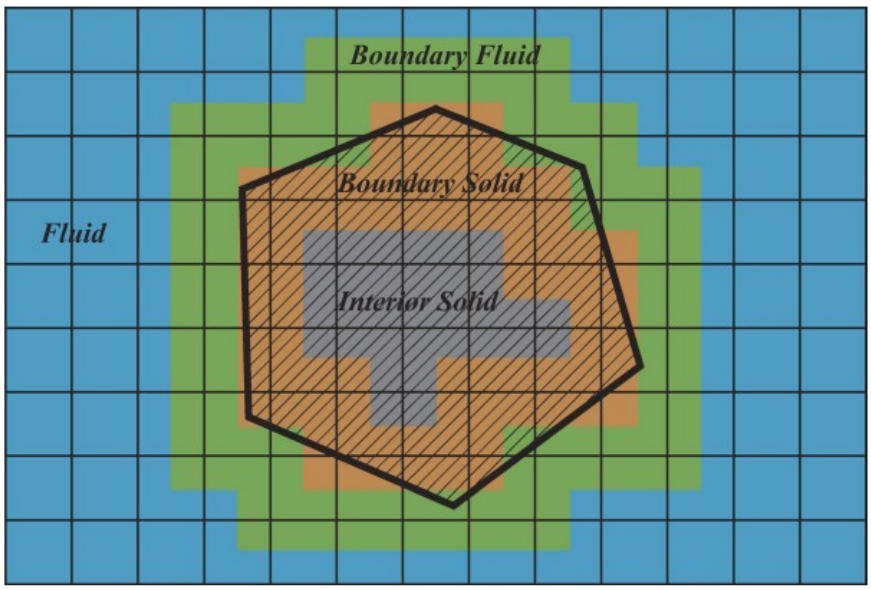


Fig. 4 Polyhedral block moving through fluid mesh. Background grid indicates lattice nodes which are at the center of fluid cells. Fluid cells are pure fluid, interior solid cells are pure solid. Boundary fluid and boundary solid cells are some proportion of fluid and solid

The partially saturated method (Noble and Torczynski 1998), here referred 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 block moves through the fluid mesh, it may partially or completely cover fluid cells, as shown in Fig. 4. The LBE is modified to accommodate the solid phase by introducing an additional solid collision operator:

$$f_i\left(\boldsymbol{x}+\boldsymbol{c}_i\Delta t, t+\Delta t\right) = f_i(\boldsymbol{x},t) + \left[1-\sum_s B(\varepsilon_s,\Omega^{\text{fluid}})\right]\Omega_i^{\text{fluid}} + \sum_s B(\varepsilon_s,\Omega^{\text{fluid}})\Omega_i^{\text{solid}}, \tag{15}$$

where ε_s is the volumetric solid fraction for each block intersecting the fluid node and $B(\varepsilon_s,\Omega^{\text{fluid}})$ is a weighting function. $B(\varepsilon_s,\Omega^{\text{fluid}})$ ranges from 0 (pure fluid) to 1 (pure solid). When $B(\varepsilon_s,\Omega^{\text{fluid}})=0$ the standard LBE is recovered, while for $B(\varepsilon_s,\Omega^{\text{fluid}})=1$ only the solid collision operator participates in the collision step. The collision operator for solid nodes is

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

where $\boldsymbol{u}_s$ is the velocity of the solid block at time $t+\Delta t$ at the fluid node. This form of Ω_i^{solid} is based on the bounce-back of the non-equilibrium portion of the particle distributions (Zou and He 1997) while the weighting function depends on the volumetric solid fraction and the fluid collision operator

parameters. For the D3Q27 MRT collision operator used, the weighting function is expressed as

$$B = \frac{\varepsilon_s\left(\frac{1}{s_5} - \frac{1}{2}\right)}{(1-\varepsilon_s) + \left(\frac{1}{s_5} - \frac{1}{2}\right)} = \frac{\varepsilon_s\left(\frac{1}{s_7} - \frac{1}{2}\right)}{(1-\varepsilon_s) + \left(\frac{1}{s_7} - \frac{1}{2}\right)}. \tag{17}$$

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

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

where x_n are all the lattice nodes that are interacting with the block and x_{CM} is the location of the center of mass of the block. 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 Fig. 4—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. We compute the volumetric solid content for the polyhedral rock blocks moving through the fluid mesh analytically using linear programming and simplex integration, as described below.

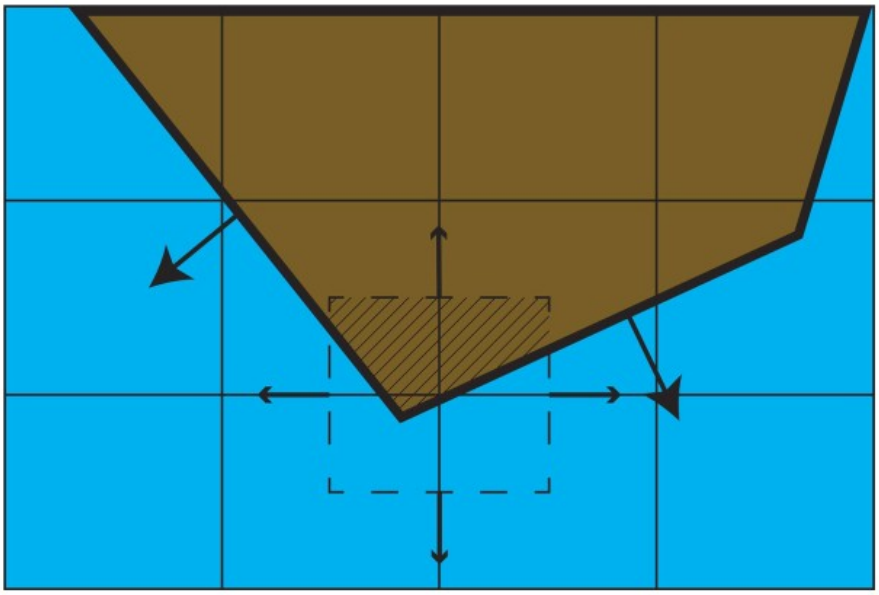


Fig. 5 Closeup of polyhedral particle overlapping fluid cell. The fluid cell is described by the normals to the cell faces and their distance from the lattice node at the center of the cell. The hatched region is where the particle and fluid cell overlap—this is the solid content of the fluid cell

When a block overlaps with a fluid cell, as shown in Fig. 5, a fraction of the volume in the cell will be occupied by the solid block. To calculate what this fraction is, it is first necessary to establish whether the fluid cell and block overlap. This is identical to the contact detection problem in DEM and is conveniently solved using the same linear programming algorithm as described in Sect. 2.1:

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

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 $s<-\varepsilon$, where ε is a specified numerical tolerance. If the fluid cell and block do not overlap, the volumetric solid content is 0. If they do overlap, it is first necessary to establish what volume of the fluid cell is covered by the solid particle. As shown in Fig. 5, this region of overlap is fully described by a subset of the faces N_S+N_F. This subset of faces can be identified by checking all faces of the block and fluid cell for redundancy. To check a particular face $n^T\boldsymbol{x}\leq d$ for redundancy in the set N_S+N_F, the following linear program is solved:

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

The face is not redundant if $|\boldsymbol{n}^T \boldsymbol{x} - d| < \varepsilon$. With this minimal set of faces, the volume fraction of solid can be calculated through simplex integration (Shi 1997):

$$
\begin{aligned}
V_{\text{overlap}} &= \int\int\int_V \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \\
&= \sum_{i=1}^{s}\sum_{k=1}^{n(i)} S_{P_0P_1^iP_k^iP_{k+1}^i}(0,0,0) \\
&= \frac{1}{6}\sum_{i=1}^{s}\sum_{k=1}^{n(i)} \begin{vmatrix} x_1^i & y_1^i & z_1^i \\ x_k^i & y_k^i & z_k^i \\ x_{k+1}^i & y_{k+1}^i & z_{k+1}^i. \end{vmatrix}
\end{aligned} \tag{21}
$$

Equation 21 describes the summation of the volumes of tetrahedra S_i that together form the three-dimensional particle. The vertices $P_1,\ldots,P_n$ describing each of the faces of the particle should be oriented counterclockwise relative to the outward normal of the face and be specified relative to a local origin P_0. In this case, P_0 is set to the location of the center of the fluid cell in question. The volumetric solid content, ε_s, is the ratio of the volume of overlap, V_{overlap}, and volume of the fluid cell. The value of the weighting function, B, can then be calculated using Eq. (17) with ε_s.

5 Implementation Validation and Performance Evaluation

The discrete element method and the LBM were implemented in C++ and coupled as described in Sect. 4. Beyond simple unit tests, integration testing was performed to verify if the implementation is able to match analytical solutions as well as produce observed response compared to experimental data. Since the resolution needed to resolve the fluid pressures acting on individual blocks requires many more fluid nodes than solid particles, in the present implementation only the fluid computations have been accelerated using parallel computing. The parallelization was performed using Kokkos (Edwards et al. 2014), a C++ library designed for performance portability. By abstracting both parallel execution and memory layout, the same source code can be compiled to target a particular architecture based on the machine where the program will be executed, utilizing central processing units (CPUs) or/and graphics processing units (GPUs). The LBM parameters were set to the optimized values presented in Suga et al. (2015). For the DEM, the normal and shear contact stiffness were 60.0 GPa and 30.0 GPa, respectively. The specific gravity of the solid particles was set to 2.65 while the damping factor was set to 0.0%.

5.1 Validation

This section provides comparisons of the numerical solutions given by our implementation of the coupled DEM–LBM formulation with both analytical and experimental data, as shown below.

5.1.1 Sliding Block Analysis

For polyhedra, the simplest analytical test case is that of a block sliding down an inclined plane. This tests the ability of the implementation to

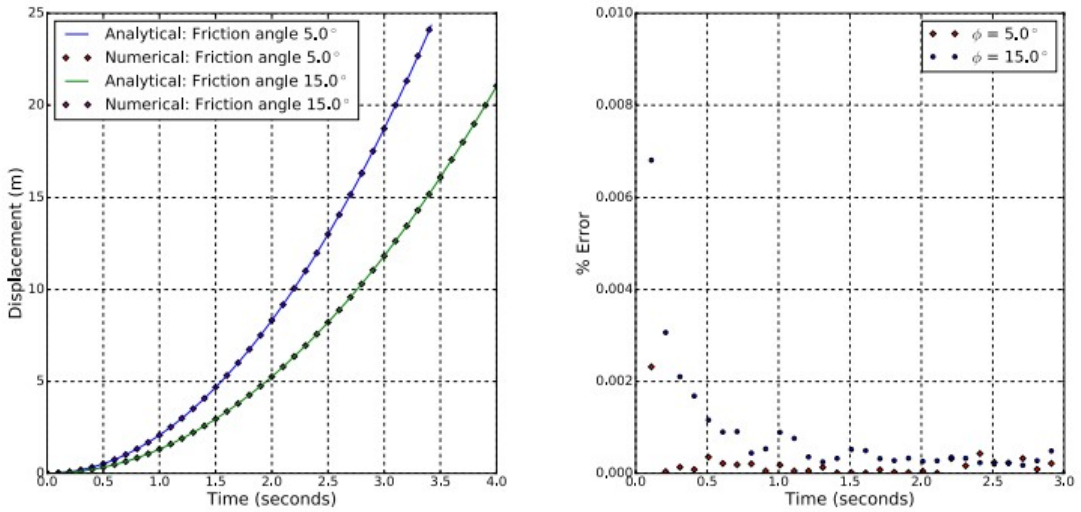
properly capture the friction force and the gravitational acceleration. For a given gravitational force, the inclination of the plane and the angle of friction between the block and sliding surface will dictate whether the block slides down the plane and, if it does, at what rate it will accelerate. In the case where sliding does occur, the position of the block on the plane is given by

$$x(t) = x_0 + v_0 t + \frac{1}{2} a t^2, \tag{22}$$

where x is the position of the block at time t, x_0 and v_0 are the initial position and velocity of the block and a is the constant gravitational force acting on the block. The acceleration is given as a function of the slope and friction angles by

$$a = (\sin(\theta) - \tan(\phi)\cos(\theta))\, g, \tag{23}$$

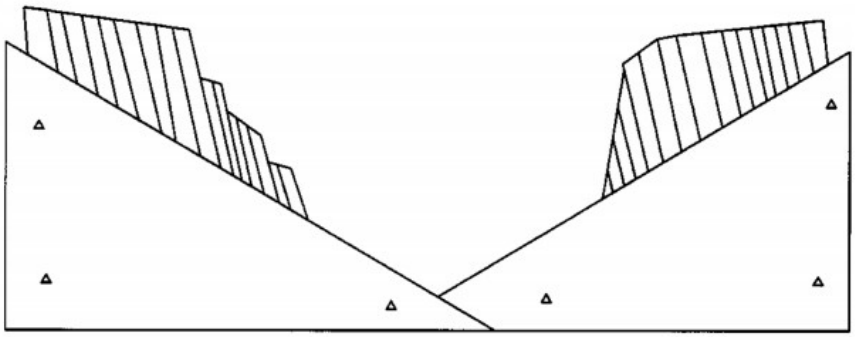
where θ is the slope angle and ϕ is the angle of friction between the block and the sliding plane.



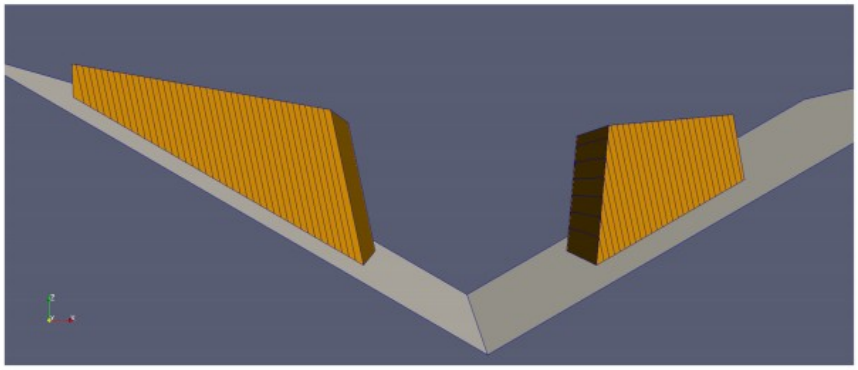
(a) Block Displacement vs Time **(b)** Percent Error vs Time

Fig. 6 Comparison of numerical results with analytical solution for sliding block

Fig. 7 Initial configuration for failure mode analysis



(a) Initial configuration from Sitar et al. (2005)



(b) Initial configuration for this analysis

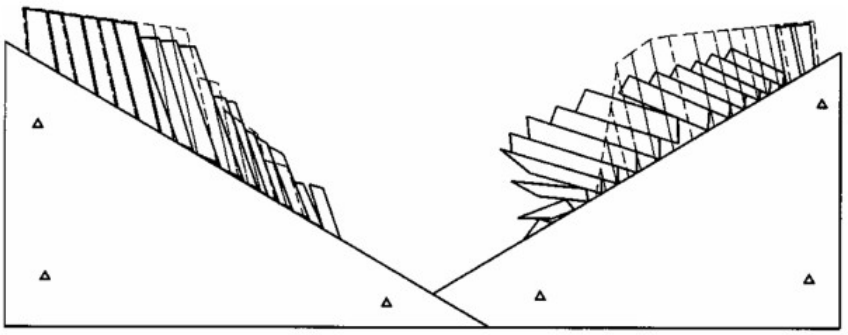
Figure 6 shows the comparison of the numerical results with the analytical solution. The numerical results match the analytical solution very well. The case where the slope angle and friction angle match was also tested—in this case no sliding should occur—and sliding did not occur in the numerical analysis. All these tests used a linear elastic contact model with Coulomb friction for the contact between the block and sliding plane.

5.1.2 Toppling and Slumping Slope Failure

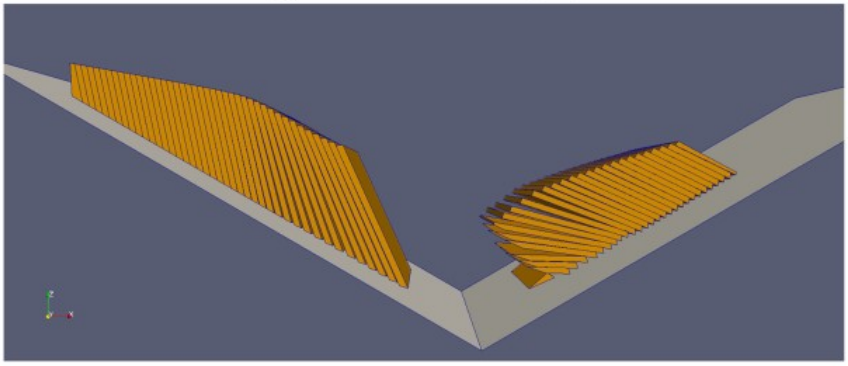
The orientation of fractures and discontinuities within rock relative to slope configuration govern how failure modes develop (Goodman and Kieffer 2000). Any discontinuous numerical model needs to be able to capture this phenomenon if it is to be used for analyzing fractured rock. To ensure our implementation of DEM functions correctly, we compared the results from a three-dimensional DEM analysis with that of a two-dimensional discontinuous deformation analysis (DDA) (Sitar et al. 2005) that illustrates how geometry of discontinuities affects kinematic behavior of rock slopes.

In this analysis, a valley or cut in pervasively jointed rock is shown to have two different failure modes on either side of the valley—the failure mode is governed by the orientation of the joints relative to the failure plane. Figure 7 shows the initial configuration from Sitar et al. (2005) alongside the initial configuration used in this analysis. Through-going joints dip at 30°, forming the sliding plane on either side of the valley. The friction angle along the sliding plane is 32° and the friction angle between the individual rock blocks

is 22°. If the rock mass was considered as a single wedge on either side of the valley, the analysis would conclude that both slopes are safe against sliding. However, as shown in Fig. 8, the slope on the right fails by toppling while the slope on the left fails by slumping (Sitar et al. 2005). Both of these modes are captured in a single analysis, illustrating the capability of DEM to capture the kinematics governing failure for different slope configurations. This is an essential feature for analyzing more complex landslides where different failure modes may initiate as the slope failure progresses.



(a) Failure modes from Sitar et al. (2005)



(b) Failure modes for this analysis

Fig. 8 Slumping (left) and toppling (right) failure modes

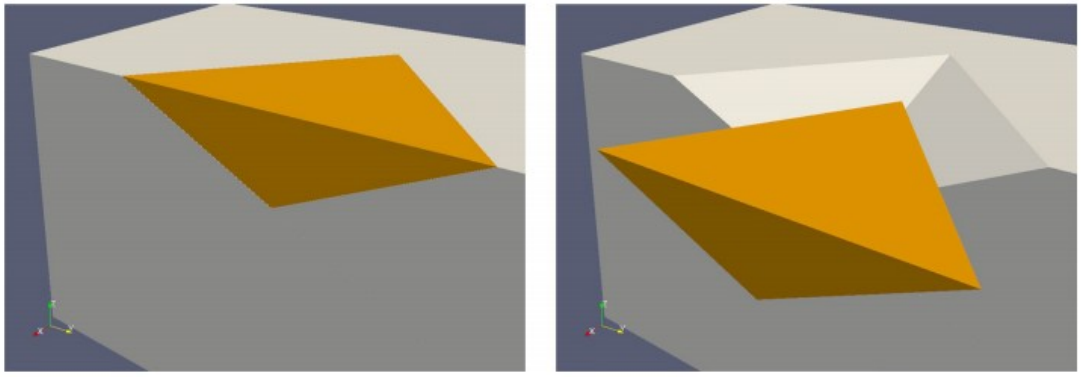


Fig. 9 Blocky rock wedge failing in sliding. Block translates along line of intersection of sliding planes

5.1.3 Wedge Sliding

Wedge sliding is a typical failure mode in blocky rock masses. For this type of failure, as shown in Fig. 9, a rock block slides without rotation along two non-parallel planes along their line of intersection. This type of failure is typical in blocky rock masses with multiple, continuous, non-parallel joint sets (Goodman and Kieffer 2000). Block theory (Goodman and Shi 1985) offers an analytical solution for determining whether a block will fail in this mode. The force F required to stabilize a block sliding along two planes with normals $\hat{n}_i$ and $\hat{n}_j$ due to the resultant $\boldsymbol{r}$ of all forces acting on the block is given by

$$F = \frac{1}{||\hat{n}_i \times \hat{n}_j||^2} [|\boldsymbol{r} \cdot (\hat{n}_i \times \hat{n}_j)| \cdot ||\hat{n}_i \times \hat{n}_j|| - |(\boldsymbol{r} \times \hat{n}_j) \cdot (\hat{n}_i \times \hat{n}_j)| \tan \phi_i - |(\boldsymbol{r} \times \hat{n}_i) \cdot (\hat{n}_i \times \hat{n}_j)| \tan \phi_j], \tag{24}$$

where ϕ_i and ϕ_j are the angles of friction on the two sliding planes. When F is negative, the block is stable, when F is positive the block is unstable. If F is zero, the block is in equilibrium and the factor of safety is essentially 1.0. When only gravitational loading is considered, r=⟨0.0,0.0,−W⟩, where W is the weight of the block.

Table 1 Wedge sliding: comparison of numerical results and block theory

$\hat{n}_i$	$\hat{n}_j$	ϕ_i	ϕ_j	$F[N]$ (Eq. 24)	DEM result
⟨0.321, − 0.383, 0.866⟩	⟨0.321, 0.383, 0.866⟩	30.0	30.0	− 5311.0	No sliding
		19.0	19.0	− 34.3	No sliding
		18.9	18.9	9.9	Sliding
		15.0	15.0	1695.3	Sliding
		15.0	30.0	− 1807.9	No sliding
		6.5	30.0	− 64.1	No sliding
		6.0	30.0	35.9	Sliding
		3.0	30.0	632.5	Sliding

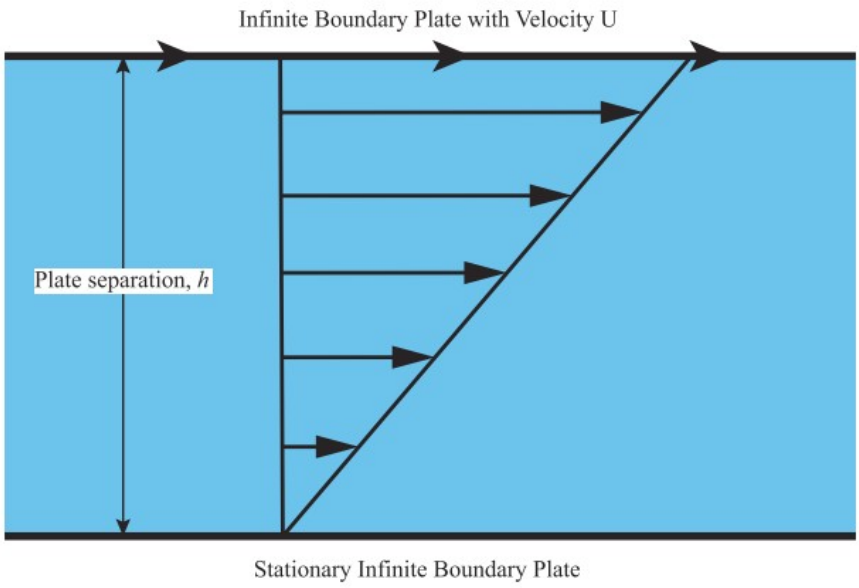


Fig. 10 Steady-state velocity profile for Couette flow with two infinite plates separated by distance h. The top plate moves with constant horizontal velocity U while the bottom plate is a stationary, no-slip boundary

To verify the ability of the DEM implementation to correctly capture the three-dimensional kinematic behavior of blocky rocks, the results from several wedge failure analyses were compared with the analytical solution from block theory. Table 1 shows the comparison of the numerical results with the predicted values from Eq. (24). The same set of joints was used for all analyses while modifying the angle of friction on the sliding planes. The numerical analysis captures the transition from stable to sliding very well and matches the results from block theory. For the cases where F was very small and negative—indicating the block is stable, but not by a great amount—the block displaced only very slightly at the beginning of the simulations. However, once enough frictional resistance was mobilized, the block stopped sliding and remained stationary for the remainder of the simulation time.

5.1.4 Couette Flow

Couette flow is a particular case of simple parallel-plate flow where one plate is fixed while the other plate moves with a constant velocity while a viscous fluid fills the space between the two plates. Figure 10 illustrates this situation and shows the steady-state solution. Couette flow demonstrates shear-driven fluid motion and, by modeling this numerically, the ability of the LBM implementation to correctly capture the viscous behavior of fluid.

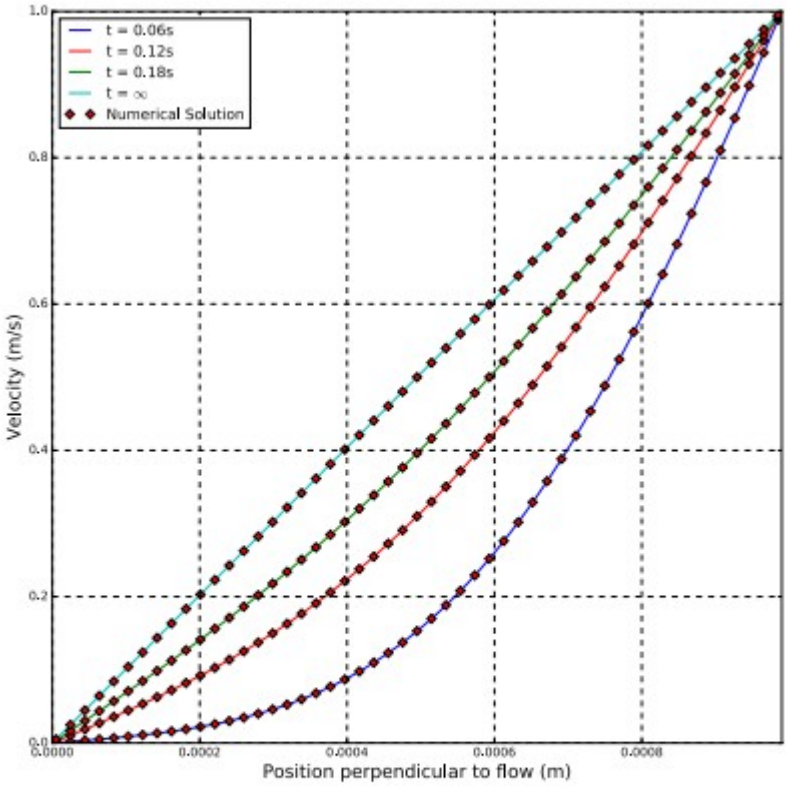


Fig. 11 Comparison of numerical results with analytical solution for Couette flow. Solid lines indicate analytical solution at different times while red diamonds show numerical results at corresponding times

The PDE describing the evolution of velocity between the two parallel plates is (Batchelor 2000)

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2} \tag{25}$$

with initial condition and boundary conditions

$$\begin{aligned} u(y,0) &= 0, \quad 0 < y < h \\ u(0,t) &= 0, \quad t \geq 0 \\ u(h,t) &= U, \quad t \geq 0. \end{aligned} \tag{26}$$

The solution to this PDE is given by Lei et al. (2016)

$$u(x,t) = -\frac{2U}{h} \sum_{n=1}^{\infty} \frac{h}{n\pi} e^{-\nu\left(\frac{n\pi}{h}\right)^2 t} \sin\left(\frac{n\pi}{h} y\right) + \frac{U}{h}(h-y). \tag{27}$$

Figure 11 shows the comparison of numerical results with the analytical solution at different times. The numerical results match the analytical

solution very well and illustrate the ability of the LBM implementation to capture shear-driven flow correctly.

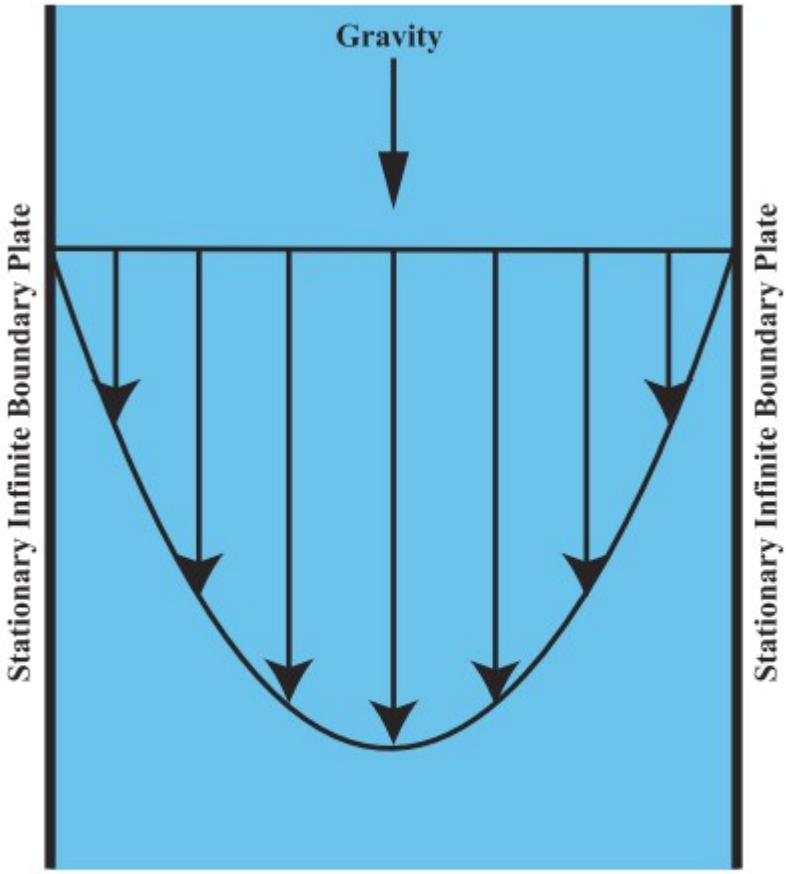


Fig. 12 Steady-state velocity profile for gravity-driven plane Poiseuille flow with two infinite plates separated by distance h. Both boundary plates are stationary, no-slip boundary conditions

5.1.5 Gravity-Driven Plane Poiseuille Flow

The validity of the body force implementation is tested by modeling gravity-driven plane Poiseuille flow. For this type of flow, two infinite plates are separated by a constant distance, as shown in Fig. 12, with a body force gg acting on the fluid. The PDE describing this type of flow is given by

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2} + g, \tag{28}$$

with initial condition and boundary conditions

$$
\begin{aligned}
&u(y,0) = 0, \quad 0 < y < h \\
&u(0,t) = 0, \quad t \geq 0 \\
&u(h,t) = 0, \quad t \geq 0.
\end{aligned}
\tag{29}
$$

The solution for gravity-driven plane Poiseuille flow is given by (Fukuchi 2011)

$$
u^*(y^*,t^*) = \frac{P^*}{2}\left[(1-y^{*2}) - 4\sum_{n=0}^{\infty}(-1)^n \frac{cos\left(\pi y^*\left(n+\frac{1}{2}\right)\right)}{\pi^3\left(n+\frac{1}{2}\right)^3} e^{-\pi^2 t^*\left(n+\frac{1}{2}\right)^2}\right], \tag{30}
$$

where

$$
t^* = \frac{t\nu}{h^2}, \quad u^* = \frac{u}{U}, \quad y^* = \frac{y}{h}, \quad P^* = \frac{gh^2}{\nu U}. \tag{31}
$$

The comparison between the analytical solution and numerical results is shown in Fig. 13. This shows excellent agreement between the numerical results and the analytical solution at various stages as the flow evolves.

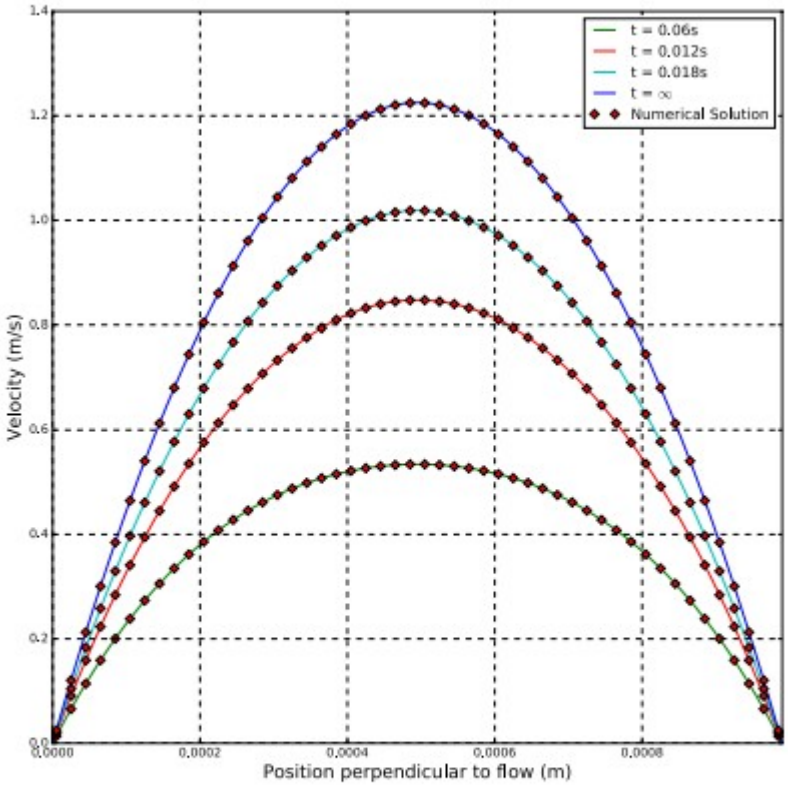


Fig. 13 Comparison of numerical results with analytical solution for gravity-driven plane Poiseuille flow. Solid lines indicate analytical solution at different times while red diamonds show numerical results at corresponding times

5.1.6 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 (Frizell 2007). The stagnation pressure and flow patterns associated with offset joints were modeled using the coupled DEM–LBM implementation to verify its ability to reproduce experimental observations. For these simulations, 1/2, 1/4, 1/8, 1/16, and 1/32 in. joints with 1/8 in. offset between slabs were modeled, similar to the physical and numerical experiments performed by Frizell (2007). Figure 14 shows the domain for the simulations. The two slabs are separated in the center of the domain by a 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 of 50 ft/s and flows toward the right side of the domain where flow is forced upward by the offset joint.

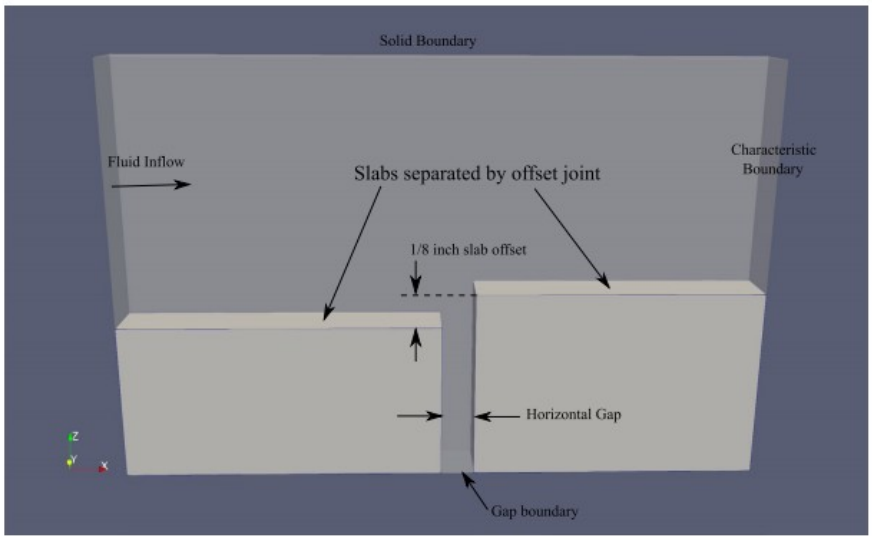
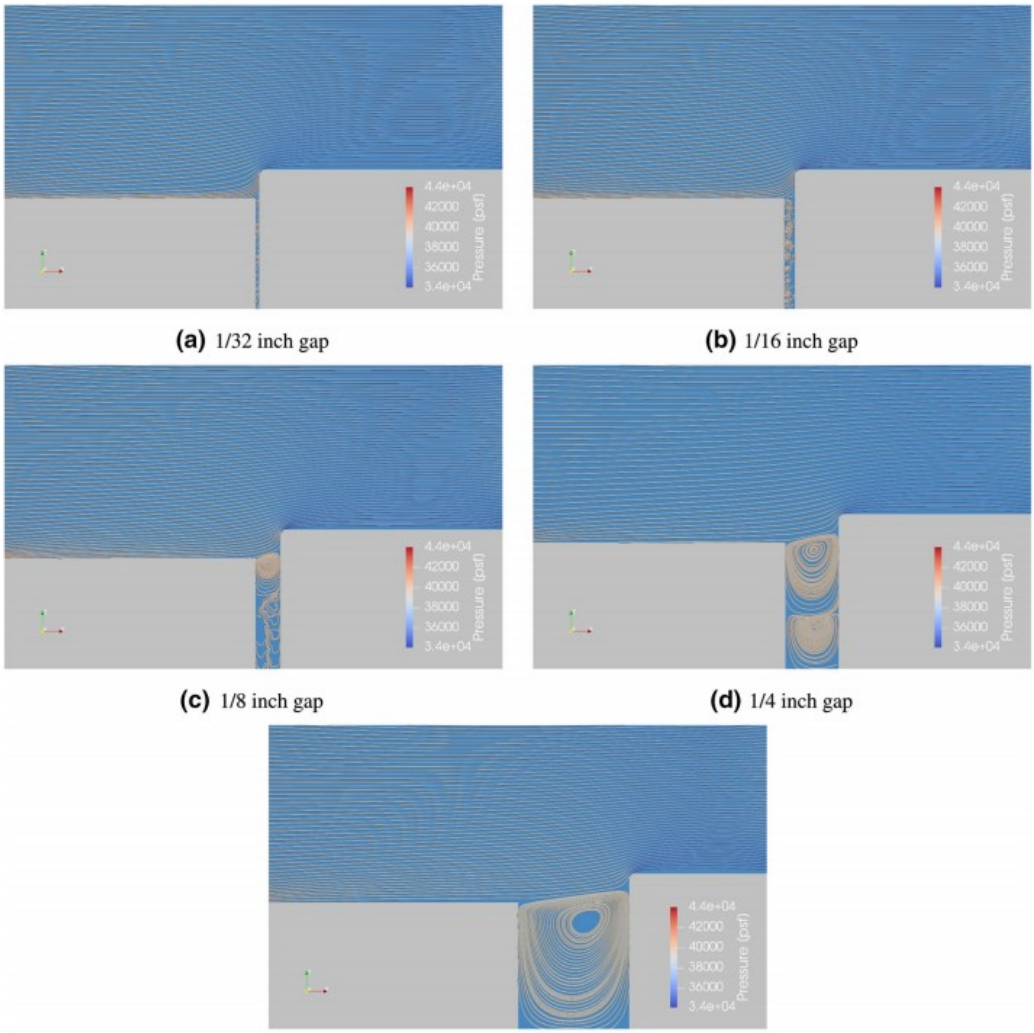


Fig. 14 Two slabs offset by 1/8 in. separated by a horizontal gap. The gap boundary is modeled as a solid boundary in the sealed case and as a characteristic boundary in the open case. Boundaries perpendicular to the *y*-axis are periodic

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. For sealed case, the uplift pressure is observed to decrease as the horizontal gap between the two slabs is increased. This observation is consistent with particle image velocimetry (PIV) measurements from the lab experiments conducted by Frizell (2007), where it was postulated that the decrease in pressure may be caused by the formation of a driven recirculation zone at the point of the gap entrance. Figure 15 shows stream tracers colored by the pressure for all the joint gaps modeled. Here, it can be seen how a zone of recirculation starts to form as the horizontal gap increases, blocking transmission of the full stagnation pressure in the gap between the slabs. These results are in excellent agreement with the experimental observations presented by Frizell (2007).



(a) 1/32 inch gap

(b) 1/16 inch gap

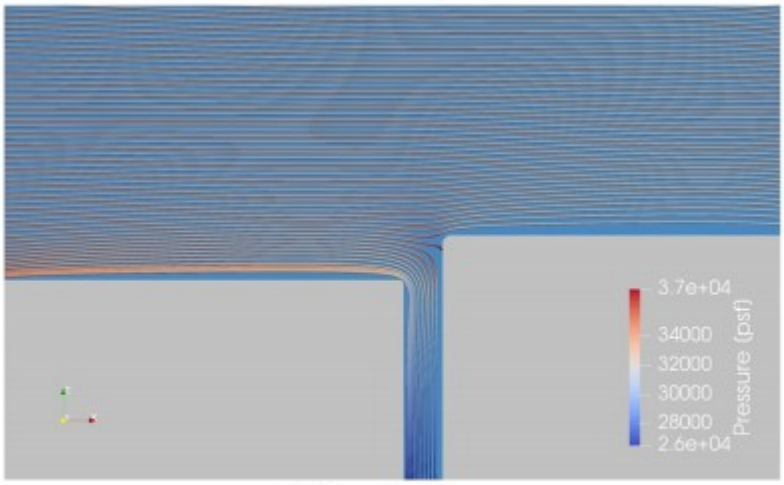
(c) 1/8 inch gap

(d) 1/4 inch gap

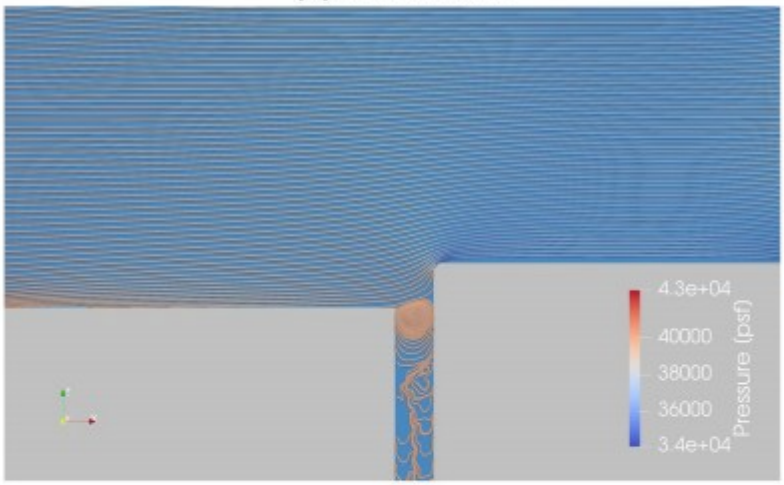
(e) 1/2 inch gap

Fig. 15 Stream tracers colored by pressure for sealed gap. As horizontal gap widens, a recirculation zone develops at the gap entrance leading to lower uplift pressures within the gap. Pressures are shown in pounds per square foot (color figure online)

Comparison between the open and sealed cases shows a decrease in the uplift pressure for the open compared to the sealed case, as shown in Fig. 16 for a horizontal gap of 1/8 in. The decrease in uplift pressure is consistent with the characteristics presented by Frizell (2007) where it was noted that both the geometry and the drain system intercepting flow through the gap will affect the uplift pressure. In the case of our simulations, the boundary is modeled using a characteristic boundary that allows fluid to exit the system while effectively eliminating returning flow. This is consistent with the experimental setup where gap geometry controlled the flow into the gap.



(a) 1/8 inch open gap



(b) 1/8 inch sealed gap

Fig. 16 Comparison of flow pattern and pressure for open and sealed cases. Uplift pressure is higher in sealed case as velocity is converted to pressure. Pressures are shown in pounds per square foot

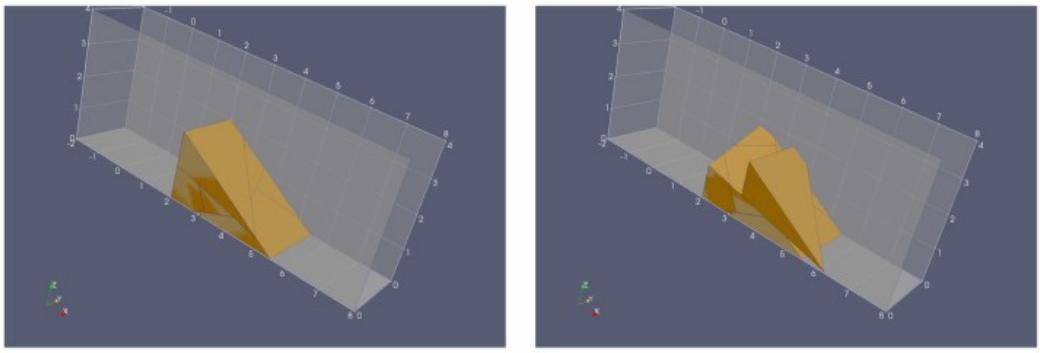
5.2 Performance

The general form of the coupled DEM–LBM formulation has potential application for analysis of a wide range of problems involving fluid–solid interaction—from wind loading on buildings to scour of bridge foundations. Herein we focus on providing illustrative example applications to rock engineering, specifically rock erosion and transport of polyhedral particles. In that context, we address the computational demand for various classes of applications and on identifying areas where future improvement is warranted or needed.

5.2.1 Rock Erosion

The principal motivation for this research was the development of capability to simulate the erosion of individual rock blocks from a fractured rock mass. In these simulations, the fractured rock mass was generated using a parallel 3-D block cutting algorithm implemented in SparkRocks (Gardner et al. 2017) with the joint set data from an unlined dam spillway in the Sierra Nevada in Northern California. This joint set data, along with the flow characteristics, are based on lab and field experiments investigating rock scour conducted by George & Sitar (George 2015; George et al. 2015).

First, the dry condition was tested to ensure that the rock mass was stable prior to any interaction with water. Figure 17a shows the initial configuration of the fractured rock with one joint set dipping unfavorably downslope. The block on the top of this unfavorable joint was unstable and failed in sliding, as shown in Fig. 17b, and it was removed prior to running analyses with fluid. The remaining rock mass was stable in the dry condition. The bottom and side boundaries were modeled as rigid, frictional boundaries.

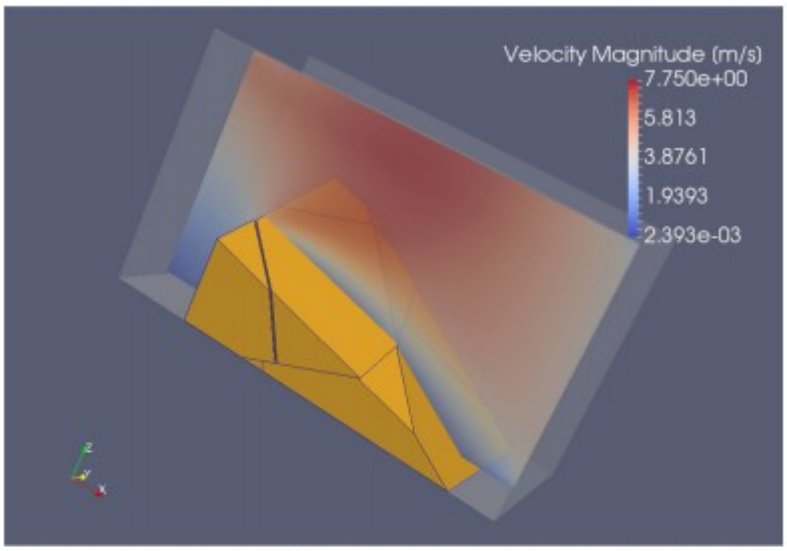


(a) Initial configuration of dry rock mass

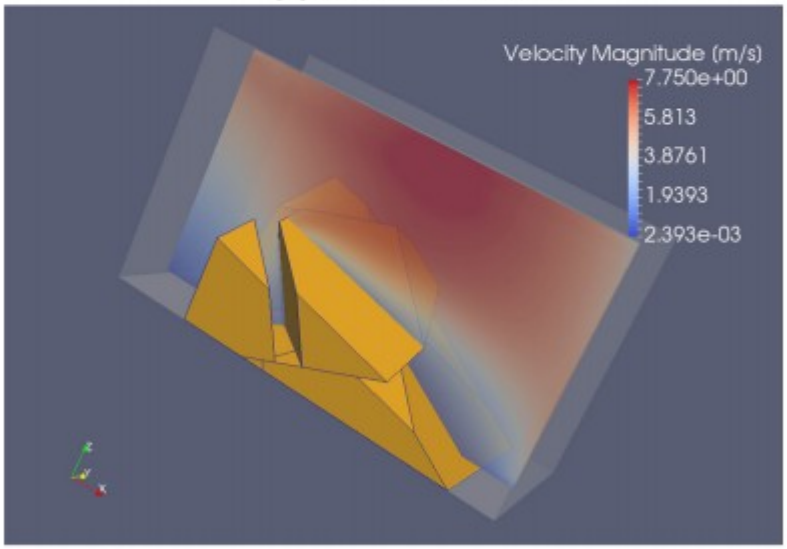
(b) Block sliding along unfavorable discontinuity in dry condition

Fig. 17 Fractured rock mass generated based on joint set data from unlined dam spillway in the Sierra Nevada in Northern California. Note: axis units are in meters

The coupled fluid–solid simulation was then performed to investigate the potential for further erosion due to hydrodynamic loading. The upstream boundary was held to a constant velocity varying linearly from 0 m/s at the bottom of the channel to 9 m/s at the top of the domain. The downstream boundary was modeled as an outlet using a non-reflecting characteristic boundary. The fluid in the remainder of the domain was initially at rest and was allowed to come to equilibrium due to the velocity boundary on the upstream side and gravitational acceleration as the water flowed downslope. During the course of the simulation, 6.0 s model time, one additional block was eroded due to hydrodynamic loading. Figure 18 shows the displacement of the block as well as the velocity magnitude along a section of the incline. Figure 19 shows a closer view of the opening fractures as the block displaces. The velocity vectors in this figure illustrate how flow through fractures increases as they open up.

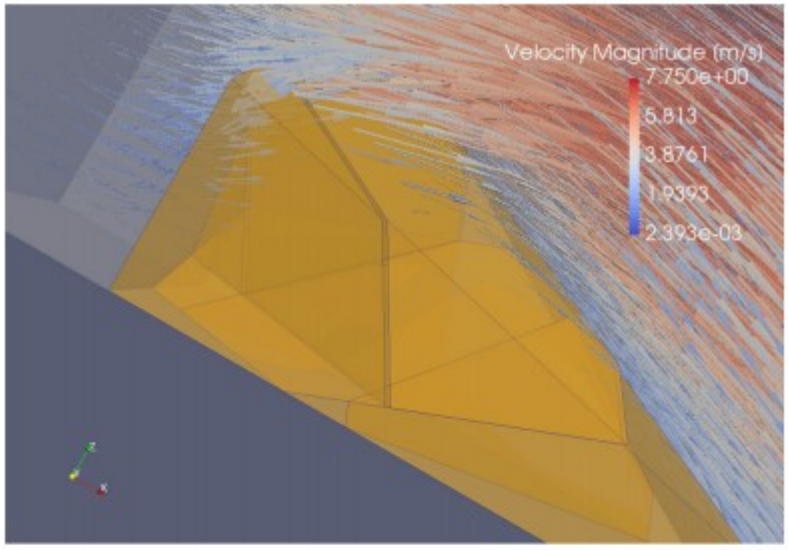


(a) Initial fracture opening

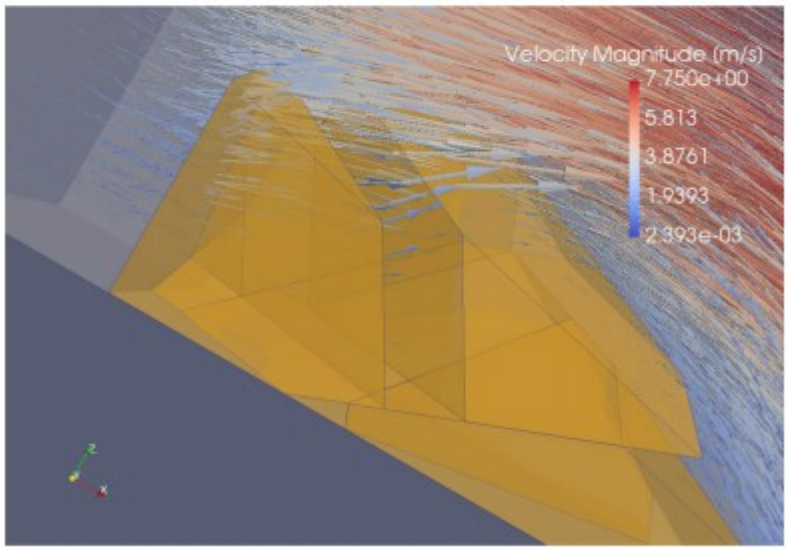


(b) Fractures opening increases as block is eroded by water

Fig. 18 Rock block sliding on fracture plane due to hydrodynamic loading



(a) Little to no flow inside fractures during initial displacement



(b) Large amount of fluid flowing through fractures as they widen

Fig. 19 Closer view of fractures and internal rock mass structure during erosion

This example problem illustrates the ability of the coupled LBM–DEM program to capture hydrodynamic loading on the fractured rock as it flows over and against the rock. However, it also highlights a significant challenge in modeling this class of problems—the difference in the scales required to capture flow both over the rock mass and through the rock mass. Only once the block has sufficiently displaced does water begin to flow through the fractures. This is a numerical issue as water most definitely flows into and through the fractures long before they have dilated as much as seen in the numerical simulations.

Numerically, in the DEM formulation blocks are allowed a slight overlap, hence resulting in completely closed fractures. A partial solution to this numerical issue is to define a “virtual fracture aperture” where the block faces are set back by some amount such that the effective sizes of the blocks manifested in the fluid mesh are slightly smaller and the fluid mesh is fine enough for a sufficient number of nodes to be present in the virtual fracture. This setback can be adjusted to match the effective fracture aperture in the rock mass based on observed flow. This approach is effective in the current implementation of the method only when the simulation domain is small enough that gross mesh refinement is able to capture the scale of the fractures before the memory demands for such a fine mesh exceed available hardware capacity.

The computational bottleneck in this type of simulation is the fluid–solid coupling computations—each of the fluid nodes identified as a being on the fluid–solid boundary needs to be checked for volumetric solid content. In the current code implementation the coupling computations are still computed in serial while the rest of the fluid computations are executed in parallel. As the fluid mesh is refined or the size or number of the blocks is increased, the number of fluid cells that need to be checked for volumetric solid content increases rapidly. As a result, the coupling computations may take anywhere from 30 to 60% of the total computation time depending on the problem configuration.

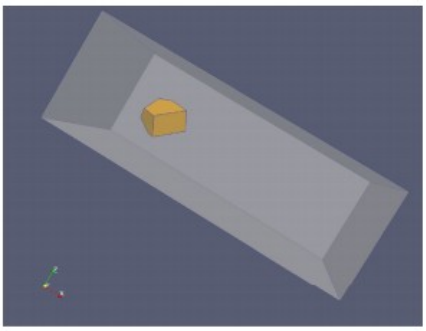
5.2.2 Transport of Polyhedral Particles

A different type of simulation problem is the transport of polyhedral particles, such as sand transport and the movement of particles along a stream bed. In these simulations, a polyhedral block is dropped onto an inclined plane and moves down the plane under the influence of gravity considering three different conditions. First, the block was allowed to roll down the plane without any water present—essentially modeling a rock fall. Next, the block started from the same initial position, but it is surrounded by stationary fluid. Lastly, the block was again dropped from its initial position, but this time within a fast-moving fluid flowing down the inclined plane with a constant inlet velocity and an outflow boundary at the bottom of the slope.

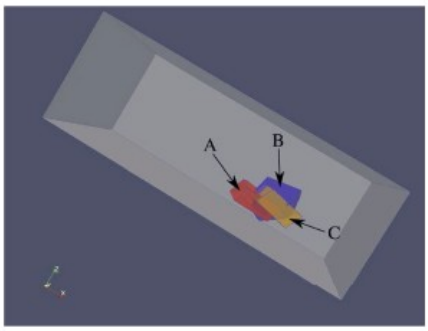
Figure 20a shows the initial block position and Fig. 20b shows a comparison of the block displacements for the different cases. The coupling correctly captures the interaction between the block and fluid: the block in stationary fluid rolls down the slope at a slower rate compared to the dry case duo to buoyancy while the fast-moving fluid carries the block downslope at a faster rate than the block in stationary fluid but it does not bounce as high as the dry case. Figure 20c shows the block displacements over time for all three cases. The inflection points in the displacement plot occur when the particles impact and bounce off of the lower boundary. Here, the buoyant effect of the water can be clearly seen through the offset in impact times for the three

different cases. Figure 21 shows how stream tracers bend round the polyhedral block as it rolls and is pushed down slope by the fluid.

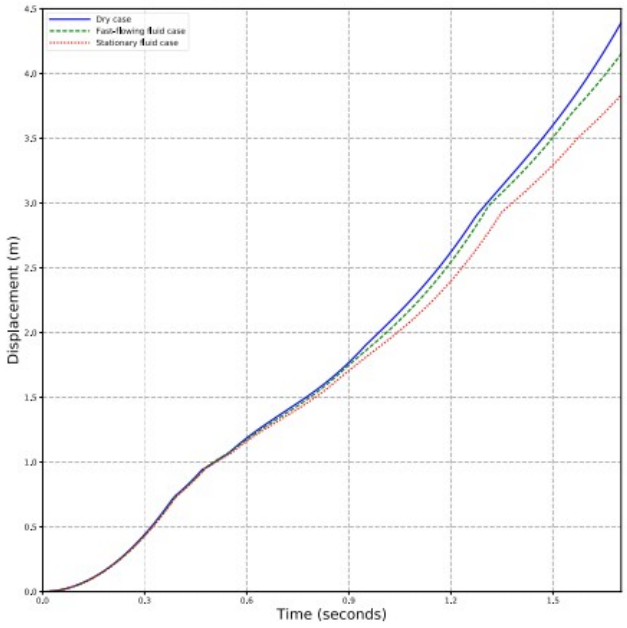
Fig. 20 Polyhedral block rolling down inclined plane



(a) Initial position of the polyhedral block for all test cases



(b) Comparison of block positions. The block labeled B is the dry case, the block labeled C is the fast-flowing fluid case and the block labeled A is the stationary fluid case. In all cases the block continues rolling down the incline.



(c) Comparison of block displacements over time

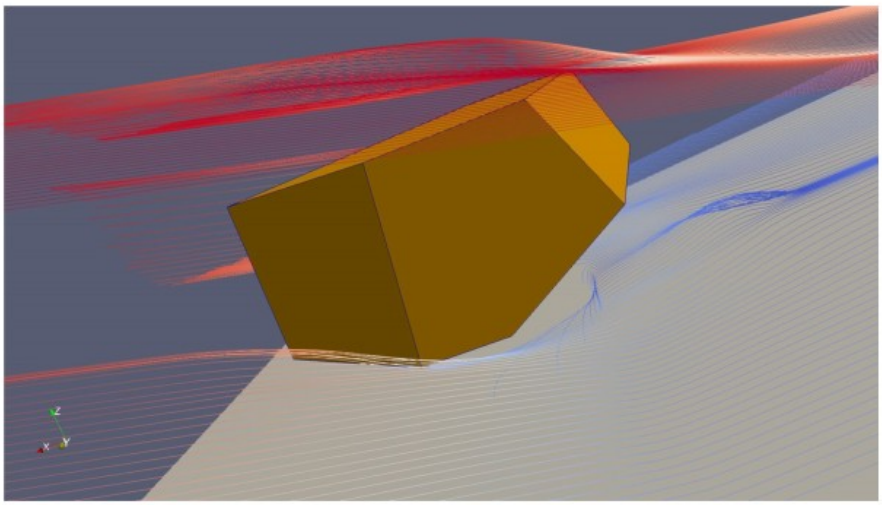


Fig. 21 Stream tracers bending round the polyhedral block moving through the fluid mesh

5.3 Computational Considerations

Table 2 contains relevant details about the hardware and simulation configurations for examples presented herein. Computationally it is important to consider the simulation time and mesh density needed for the simulations. 726,291 fluid nodes were used in the flume model in Sect. 5.2.2 while 33,201 fluid nodes were used in the rock erosion problem presented in Sect. 5.2.1. The computation time required for 0.1 s of model time for the block rolling down the incline was approximately 10 min while it took approximately 63 min of computation time to simulate 0.1 s for the rock erosion example on a computer with 2 Intel Xeon E5-2630 CPUs (6 cores each) and 20 GB of memory. This difference in computation time illustrates the computational demand of the fluid–solid coupling computations and the need to accelerate them in the source code. For the block rolling down the incline, doubling the size of the domain increases the total computation time less than if the domain size was held constant but the mesh refined by one order of magnitude. For these two cases the number of nodes is the same, but the number of fluid–solid boundary nodes is greater in the case where the mesh is refined. The computation time increases by a factor of approximately 1.4 for the increased domain size while it increases by a factor of 4.63 for the refined mesh case, though the number of nodes in the simulations is exactly the same. This illustrates that the bottleneck in the fluid–solid coupling computations is exacerbated in cases when a greater fraction of the fluid nodes interact with solids as opposed to just having more fluid nodes overall.

Table 2 Example analysis configurations

Simulation	Number of nodes	Computation time per model time	Solid fraction in fluid mesh	Moving/fixed solid?
Rock erosion	33,201	43,200	Very high	Moving
Transport of polyhedral particles	726,291	5700	Low	Moving
	5,704,581	8570	Very low	Moving
	5,704,581	27,770	High	Moving
Uplift forces	200,200	17,558	High	Fixed

All simulations were executed on a machine with two Intel Xeon E-5-2630 CPUs (12 cores, 24 threads) with 20 GB of memory

For the analysis presented in Sect. 5.1.6, the volumetric solid fraction only needs to be calculated once at the beginning of the simulation since the slabs are kept stationary during the analysis. This greatly improves the performance since the volumetric solid does not need to be evaluated at every time step. For the same size time step and approximately the same number of nodes, simulating 0.1 s of model time can be completed roughly 20 times faster for these analyses than the simulations of a block rolling down an incline described in Sect. 5.2.2.

6 Conclusion

The objective of this research is the development of tools for the evaluation and modeling of the potential for rock erosion and rock transport by fast flowing water in rock channels as these processes have been shown to have a significant impact on the performance of unlined dam spillways and other structures. The natural jointing within the rock leads to fractured rock masses comprised of irregularly shaped polyhedral blocks. The orientation of the blocks relative to slope geometry has been shown to govern the kinematic response of the rock mass (Goodman and Kieffer 2000). This inherent geometric complexity and discontinuous nature of the rock mass require numerical methods that can model the discrete interactions between individual blocks. Additionally, the shape and orientation of the individual blocks within the rock mass need to be explicitly accounted for when modeling rock–water interaction—the hydrodynamic forces acting on each block need to be modeled directly—which is highly computationally intensive. Therefore, it is necessary to develop tools that can capture the kinematics of the rock mass response and that take advantage of parallel computing.

Accordingly, a new coupled DEM–LBM program was developed to analyze the interaction between a blocky rock mass and fluid. The DEM implementation is able to correctly capture the rock mass kinematics by considering the shape and orientation of each individual block. The contact detection algorithm used for the DEM computations is based on a linear programming approach (Boon et al. 2012) which is able to consider three-dimensional polyhedral blocks. The coupling between DEM and the weakly compressible LBM is achieved through application of a volume-fraction approach that is able to consider three-dimensional polyhedral blocks interacting with fluid

(Gardner and Sitar 2018). Additionally, the fluid computations are accelerated using the C++ library Kokkos (Edwards et al. 2014) such that computations can be executed on both the CPU and GPU.

The capabilities of the coupled DEM–LBM implementation were explored by evaluating the performance characteristics of the program in modeling different types of problems involving solid–fluid interaction. The examples illustrate that the implementation is capable of simulating the complex interaction between rock and water. However, the capability to model full-scale problems is still elusive due to the enormous computational effort dictated by the multiscale nature of such problems. Therefore, significant future effort is required to improve the computational speed and model efficiency to allow realistic representation of natural settings.

Solid–fluid interaction in rock scour is a multiscale problem and improvements to better capture this are necessary. The required mesh resolution to accurately resolve hydrodynamic forces of water flowing over and around rock blocks is significantly less than the mesh resolution required to capture the interaction between the fractured rock mass and water flowing through the discontinuities. Clearly, the size of the mesh within the fractures would have to be much finer compared to the rest of the domain and simply applying that fine of a mesh to the entire simulation domain is not feasible in terms of memory demands. Additionally, once the blocks start displacing the mesh density requirements would change as the fractures open up and the blocks begin to move through the fluid mesh. Adaptive meshing and multigrid methods would be ideal for capturing these different scales of interaction.

Lastly, the high computational and memory demands for three-dimensional direct simulation of rock–water interaction requires parallel computing and efficient use of computing resources. The fluid computations have been accelerated to use shared memory parallelism on both the CPU and GPU; however, the DEM–LBM coupling and DEM computations are currently executed in serial. Accelerating this portion of the computations would greatly improve performance and enhance the capability for simulating fluid–solid interaction at a greater scale. Though the current parallelization has been implemented using shared memory, future acceleration should also consider including distributed memory parallelism such that more computational resources can be brought to bear on increasingly large simulations.

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. Additionally, we would like to thank the anonymous reviewer for their constructive feedback and suggestions.

References

Anderson TB, Jackson R (1967) Fluid mechanical description of fluidized beds. Equations of motion. Ind Eng Chem Fundam 6(4):527–539. https://doi.org/10.1021/i160024a007

Batchelor GK (2000) An introduction to fluid dynamics. Cambridge University Press, Cambridge

Bhatnagar PL, Gross EP, Krook M (1954) A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys Rev 94:511–525. https://doi.org/10.1103/PhysRev.94.511

Boon C, Houlsby G, Utili S (2012) A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method. Comput Geotech 44:73–82. https://doi.org/10.1016/j.compgeo.2012.03.012

Cundall P (1988) 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 25(3):107–116. https://doi.org/10.1016/0148-9062(88)92293-0

Cundall PA, Strack ODL (1979) A discrete numerical model for granular assemblies. Gèotechnique 29(1):47–65. https://doi.org/10.1680/geot.1979.29.1.47

d'Humières D, Ginzburg I, Krafczyk M, Lallemand P, Luo LS (2002) Multiple-relaxation-time lattice Boltzmann models in three dimensions. Philos Trans R Soc Lond A Math Phys Eng Sci 360(1792):437–451. https://doi.org/10.1098/rsta.2001.0955. http://rsta.royalsocietypublishing.org/content/360/1792/437. http://rsta.royalsocietypublishing.org/content/360/1792/437.full.pdf

Edwards HC, Trott CR, Sunderland D (2014) Kokkos: enabling manycore performance portability through polymorphic memory access patterns. J Parallel Distrib Comput 74(12):3202–3216. https://doi.org/10.1016/j.jpdc.2014.07.003

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

Fukuchi T (2011) Numerical calculation of fully-developed laminar flows in arbitrary cross-sections using finite difference method. AIP Adv 1(4):042,109. https://doi.org/10.1063/1.3652881

Gardner M, Sitar N (2018) Coupled 3-D DEM-LBM model for simulation of dynamic rock-fluid interaction. Tech. rep., University of California, Berkeley. https://doi.org/10.13140/RG.2.2.21301.73441

Gardner M, Kolb J, Sitar N (2017) Parallel and scalable block system generation. Comput Geotech 89:168–178. https://doi.org/10.1016/j.compgeo.2017.05.001

Geier M, Schnherr M, Pasquali A, Krafczyk M (2015) The cumulant lattice Boltzmann equation in three dimensions: theory and validation. Comput Math Appl 70(4):507–547. https://doi.org/10.1016/j.camwa.2015.05.001

George MF (2015) 3D block erodibility: dynamics of rock–water interaction in rock scour, PhD thesis, copyright—database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works. https://search-proquest-com.libproxy.berkeley.edu/docview/1778844728?accountid=14496. Last updated 31 May 2016

George M, Sitar N (2016) 3D block erodibility: dynamics of rock-water interaction in rock scour. Tech. rep. https://doi.org/10.13140/RG.2.2.29909.52965

George M, Sitar N, Sklar L et al (2015) Experimental evaluation of rock erosion in spillway channels. In: 49th US rock mechanics/geomechanics symposium. American Rock Mechanics Association

Goodman RE, Kieffer DS (2000) Behavior of rock in slopes. J Geotech Geoenviron Eng 126(8):675–684. https://doi.org/10.1061/(ASCE)1090-0241(2000)126:8(675)

Goodman R, Shi G (1985) Block theory and its applications to rock engineering. Prentice-Hall, Englewood Cliffs

Guo Z, Zheng C, Shi B (2002) Discrete lattice effects on the forcing term in the lattice Boltzmann method. Phys Rev E 65(4):046,308

Hart R, Cundall P, Lemos J (1988) 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 25(3):117–125. https://doi.org/10.1016/0148-9062(88)92294-2

Holdych DJ (2003) Lattice Boltzmann methods for diffuse and mobile interfaces, Ph.D. thesis. University of Illinois at Urbana-Champaign

Houlsby G (2009) Potential Particles: a method for modelling non-circular particles in DEM. Comput Geotech 36:953–959. https://doi.org/10.1016/j.compgeo.2009.03.001

Johnson SM, Williams JR, Cook BK (2007) Quaternion-based rigid body rotation integration algorithms for use in particle methods. Int J Numer Methods Eng 74(8):1303–1313. https://doi.org/10.1002/nme.2210

Kolke D (2017) An aerial view of the damaged Oroville Dam spillway. https://www.facebook.com/CADWR/photos/a.10154476288967449.1073742016.95205192448/10154476289472449/?type=3&theater

Krüger T, Kusumaatmaja H, Kuzmin A, Shardt O, Silva G, Viggen EM (2017) The lattice Boltzmann method. Springer, Berlin

Lei C, Gang L, Guozhong Z, Yuannan T, Chai J (2016) Transient stage comparison of Couette flow under step shear stress and step velocity boundary conditions. Int Commun Heat Mass Transf 75:232–239. https://doi.org/10.1016/j.icheatmasstransfer.2015.10.011

LeVeque RJ (2002) Finite volume methods for hyperbolic problems. Cambridge University Press, Cambridge

Li H, Pan C, Miller CT (2005) Pore-scale investigation of viscous coupling effects for two-phase flow in porous media. Phys Rev E 72(026):705. https://doi.org/10.1103/PhysRevE.72.026705

Luo LS (1998) Unified theory of lattice Boltzmann models for nonideal gases. Phys Rev Lett 81:1618–1621. https://doi.org/10.1103/PhysRevLett.81.1618

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

Mikito F, Daisuke N (2014) Robust coupled fluid-particle simulation scheme in Stokes-flow regime: toward the geodynamic simulation including granular media. Geochem Geophys Geosyst 15(7):2865–2882. https://doi.org/10.1002/2014GC005281

Nicoud F, Ducros F (1999) Subgrid-scale stress modelling based on the square of the velocity gradient tensor. Flow Turbul Combust 62(3):183–200. https://doi.org/10.1023/A:1009995426001

Noble DR, Torczynski JR (1998) A lattice-Boltzmann method for partially saturated computational cells. Int J Mod Phys C 09(08):1189–1201. https://doi.org/10.1142/S0129183198001084

Owen DRJ, Leonardi CR, Feng YT (2010) An efficient framework for fluid–structure interaction using the lattice Boltzmann method and immersed moving boundaries. Int J Numer Methods Eng 87:66–95. https://doi.org/10.1002/nme.2985

Shi GH (1997) Working forum on manifold method of material analysis, vol 2. The numerical manifold method and simplex integration. Tech. rep., DTIC Document

Sitar N, MacLaughlin MM, Doolin DM (2005) Influence of kinematics on landslide mobility and failure mode. J Geotech Geoenviron Eng 131(6):716–728. https://doi.org/10.1061/(ASCE)1090-0241(2005)131:6(716)

Strack E, Cook B (2007) Three-dimensional immersed boundary conditions for moving solids in the lattice-Boltzmann method. Int J Numer Methods Fluids 55(2):103–125. https://doi.org/10.1002/fld.1437

Succi S (2001) The lattice Boltzmann equation: for fluid dynamics and beyond. Oxford University Press, Oxford

Succi S, Foti E, Higuera F (1989) Three-dimensional flows in complex geometries with the lattice Boltzmann method. Europhys Lett 10(5):433. http://stacks.iop.org/0295-5075/10/i=5/a=008

Suga K, Kuwata Y, Takashima K, Chikasue R (2015) A D3Q27 multiple-relaxation-time lattice Boltzmann method for turbulent flows. Comput Math Appl 69(6):518–529. https://doi.org/10.1016/j.camwa.2015.01.010

Swope WC, Andersen HC, Berens PH, Wilson KR (1982) 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 76(1):637–649. https://doi.org/10.1063/1.442716

Tsuji Y, Kawaguchi T, Tanaka T (1993) Discrete particle simulation of two-dimensional fluidized bed. Powder Technol 77(1):79–87. https://doi.org/10.1016/0032-5910(93)85010-7

Tsuji T, Ito A, Tanaka T (2008) Multi-scale structure of clustering particles. Powder Technol 179(3):115–125. https://doi.org/10.1016/j.powtec.2007.07.003

Williams JR, Perkins E, Cook B (2004) A contact algorithm for partitioning n arbitrary sized objects. Eng Comput 21(2/3/4):235–248. https://doi.org/10.1108/02644400410519767

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

Zienkiewicz OC, Taylor RL, Nithiarasu P (2014) The finite element method for fluid dynamics. Butterworth-Heinemann, Oxford. https://libproxy.berkeley.edu/login?qurl=http%3a%2f%2fsearch.ebscohost.com%2flogin.aspx%3fdirect%3dtrue%26db%3dcat04202a%26AN%3ducb.b22584019%26site%3deds-live

Zou Q, He X (1997) On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. Phys Fluids 9(6):1591–1598. https://doi.org/10.1063/1.869307