Title

Modeling of rock scour using coupled 3-D discrete element and lattice boltzmann methods

Permalink

https://escholarship.org/uc/item/7sp24320

Authors

Gardner, MH

Sitar, N

Publication Date

2018

Peer reviewed

Modeling of Rock Scour using Coupled 3-D Discrete Element and Lattice Boltzmann Methods

Gardner, M.H. and Sitar, N.

University of California, Berkeley, California, United States of America

This paper was prepared for presentation at the 52nd US Rock Mechanics / Geomechanics Symposium held in Seattle, Washington, USA, 17–20 June 2018. This paper was selected for presentation at the symposium by an ARMA Technical Program Committee based on a technical and critical review of the paper by a minimum of two technical reviewers. The material, as presented, does not necessarily reflect any position of ARMA, its officers, or members. Electronic reproduction, distribution, or storage of any part of this paper for commercial purposes without the written consent of ARMA is prohibited. Permission to reproduce in print is restricted to an abstract of not more than 200 words; illustrations may not be copied. The abstract must contain conspicuous acknowledgement of where and by whom the paper was presented.

ABSTRACT: Scour of rock in unlined rock spillway channels is a critical issue facing many of the world’s dams. From a modeling point of view this poses a challenging and interesting problem that combines rock mechanics and hydraulics of turbulent flow. We analyze this interaction between the blocky rock mass and water by directly modeling the solid and fluid phases—the individual polyhedral blocks are modeled using the Discrete Element Method (DEM) while the 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 of the rock surface 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. Herein we present preliminary results to demonstrate the capabilities of the methodology.

1. INTRODUCTION

Scour of rock is a very 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. The most current example of this problem is Oroville Dam in Northern California—massive scour damage to both the service and emergency spillways during the flood events of February 2017 led to the evacuation of more the 188,000 people living downstream of the dam.

In order to effectively model rock scour, it is necessary to consider the interaction between the blocky rock mass and the water flowing over and through it. Simulations modeling this process can follow one of two approaches:

- Account for fluid-solid interaction based on a locally averaged approach (Anderson & Jackson, 1967, Tsuji et al., 2008, Furuichi et al., 2014)
- Directly simulate hydrodynamic forces on the solid particles (Noble & Torczynksi, 1998, Holdych, 2003, Hölzer & Sommerfeld, 2009)

In the locally averaged approach, the number of solid particles is greater than the number of fluid cells, making this approach less computationally expensive. However, since the fluid-solid coupling is done on a volumeaveraged basis, all particles within a local region will experience the same hydrodynamic forces. In certain applications this may be appropriate, but for rock scour this approach does not offer sufficient resolution.

The second approach attempts to overcome this shortcoming by directly simulating the hydrodynamic forces on the solid particles. To achieve this, it is necessary to have a much higher resolution fluid mesh and consequently many more fluid cells compared to solid particles. This added accuracy makes direct simulations significantly more computationally expensive.

Our approach falls into the second category where fluidsolid interaction is directly simulated. This is achieved by coupling the Discrete Element Method (DEM) with the Lattice Boltzmann Method (LBM) in three dimensions. The solid polyhedral particles are modeled using DEM while the fluid phase is simulated using LBM. As previously mentioned, this approach is computationally intensive and requires parallel computing to accelerate computations. We implemented our computations in parallel using the Kokkos C++ library (Edwards et al. 2014). Kokkos achieves performance portability among different computing platforms, allowing the same code to be compiled to target the hardware it will be executed on. The resulting software, written in C++, is capable of modeling fluid-solid interaction in three dimensions and can be executed in parallel to accelerate computations.

2. SOLID PHASE MODEL

In fractured rock, the displacements occur primarily along the joints within the fractured rock mass—the three dimensional orientation of the discontinuities largely influence the block removability, kinematics and stability (Goodman & Shi, 1985). Therefore, it is important that any numerical model used to describe the mechanical behavior of rock is able to capture the discontinuous nature of the rock. Continuum models are not able to capture this behavior since the rock is not continuous at all. Distinct particle methods such as the Discrete Element Method (Cundall & Strack, 1979) and Discontinuous Deformation Analysis (Shi & Goodman, 1988) directly model individual particles, making them ideal for modeling fractured rock.

We chose to use DEM since the time integration formulation is explicit. This means that all block force and moment computations are local and, once all contact detection has been completed for a particular time step, the block displacements and rotations can be updated without needing any additional information about neighboring blocks. This local nature of the calculations makes them attractive for parallel computing.

2.1. Formulation

The equations of translational and rotational motion for an individual rock block are:

$$\ddot{x}_i + \alpha \dot{x}_i = \frac{F_i}{m} + g_i \qquad (1)$$

$$\dot{\omega}_i + \alpha \omega_i = \frac{M_i}{I} \qquad (2)$$

where $\ddot{x}_i$ and $\dot{\omega}_i$ represent the translational and rotational acceleration of the block; F_i and M_i are the total force and moment acting on the block; α is the damping constant; and g_i is the gravitational acceleration. These equations are integrated over time using a velocity Verlet time integrator for translation and a quaternion-based time integrator (Johnson et al., 2008) for rotation.

The total force and torque acting on the block are determined based on contact between blocks. Contact forces and moments are calculated following the method of Hart et al., 1988.

2.2. Contact Detection

In order to determine the contact forces and moments, it is necessary to establish which blocks are in contact with each other. This is the most computationally expensive portion of DEM—approximately 80% of computation time is spent here (Horner et al., 2000).

Establishing contact between blocks is done in two phases: first, a neighbor search to establish which blocks are close enough to possibly be in contact and, second, checking whether those blocks are actually in contact. We implemented the CGRID algorithm (Williams et al., 2003) to perform the neighbor search. Unlike other neighbor search algorithms, such as NBS (Munjiza & Andrews, 1998), CGRID is able to maintain *O(N)* operations to complete spatial binning even if the particle sizes are substantially different —often the case for fractured rock.

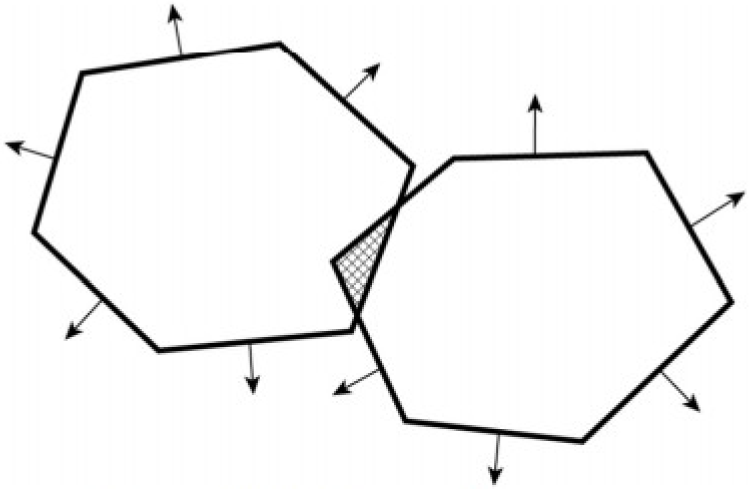


Figure 1: Two colliding block with overlapping region. Arrows indicate the direction of normal vectors to block faces (After Boon et al., 2012).

Once the neighbor search has established which blocks could possibly be in contact, the neighboring blocks are checked to determine which pairs are actually in contact. For this phase of the computations, we used a linear programming approach as described by Boon et al., 2012. As shown in Figure 1, each polyhedral block can be defined as a set of linear inequalities describing the faces of the block. Using this formulation, contact detection can be recast as a convex optimization. This algorithm greatly simplifies the contact detection process—the only data necessary for describing the blocks is the normal of each of the block faces and the distance of that face from the block centroid. Contact is established by solving a linear program; if contact exists, the contact point is taken as the analytical center of the region of overlap, indicated by the hashed region in Figure 1. The contact normal is calculated using the gradient vector of “potential particles” within each of the contacting blocks.

3. FLUID PHASE MODEL

The behavior of viscous fluid is described by the Navier Stokes equation. In computational fluid dynamics, several methods have been used to simulate the behavior of fluid by approximating the Navier Stokes equation numerically. The Finite Element Method (FEM) and Finite Volume Method (FVM) are two of the most popular methods; they can capture shocks and can offer higher order accuracy. However, when a solid phase is allowed to move through the fluid phase; these methods can become prohibitively computationally expensive and complicated: As the fluid moves through the FEM domain, it is necessary to re-mesh the fluid in the vicinity of the block and translate solutions from the old mesh to the updated one. For the FVM, establishing the support for higher order solutions is no longer clear when fluid cells are covered by solids.

An alternative to using the FEM or FVM, is the Lattice Boltzmann Method (LBM). Instead of solving the Navier Stokes equations, the LBM is based on kinetic theory and solves a mesoscopic description of fluid behavior—distributions of particles form the fundamental description of fluid behavior. It can be shown that the LBM recovers the macroscopic Navier-Stokes equations through a Chapman-Enskog expansion (Succi, 2001). The LBM formulation is intrinsically parallelizable and the method is able to deal with complex geometries and solids moving through the fluid mesh in a relatively straightforward fashion.

3.1. Formulation

In addition to discretizing physical space and time, the LBM also discretizes velocity space in the Boltzmann equation. The set of discrete velocities and accompanying weights are selected to satisfy the correct macroscopic conservation laws (Krüger et al., 2016). Figure 2 shows two such sets, one with 9 discrete velocities in two dimensions and another with 27 discrete velocities in three dimensions.

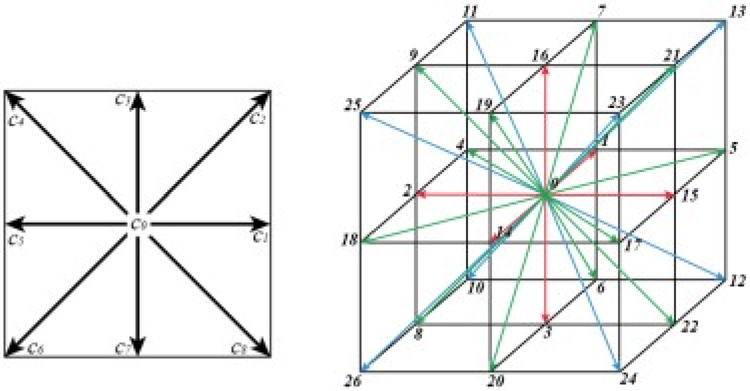


Figure 2: D2Q9 and D3Q27 discrete velocity sets

This discretization in time, physical and velocity space of the Boltzmann equation leads to the lattice Boltzmann equation:

$$f_i(x+c_i\Delta t, t+\Delta t) - f_i(x,t) = \Omega_i(x,t) \qquad (3)$$

This equation describes the two steps in the LBM: streaming and collision. In the streaming step, particle populations $f_i(x,t)$ move to neighboring point $x + c_i\Delta t$ with velocity c_i at the next time step $t + \Delta t$. The collision step is described by the collision operator, Ω_i, which models particle collisions by redistributing particles among the populations $f_i(x,t)$ at each node in the fluid mesh.

There are many different collision operators available. The simplest is the single relaxation time BhatnagarGross-Krook (BGK) operator (Bhatnagar et al., 1954) while other models offer increasingly more relaxation times to account for the difference in rates at which hydrodynamic quantities relax toward equilibrium. We implemented a three-dimensional, 27 discrete velocity, multiple relaxation time model (D3Q27 MRT) as described by Suga et al., 2015. This model offers more numerical stability and accuracy by decoupling the relaxation of different hydrodynamic quantities in moment space rather than population space, allowing them to relax to equilibrium at different rates. Once relaxation is completed, moments are transformed back to population space where the streaming step is done as described in the lattice Boltzmann equation.

A full description of the D3Q27 MRT collision operator is available in Suga et al., 2015, but it is important to mention that as with all LBM methods the spatial and temporal discretization are tied to the fluid viscosity by at least one of the relaxation times. The relation for the D3Q27 MRT model is:

$$\nu = c_s^2\left(\frac{1}{s_5}-\frac{1}{2}\right)\frac{\Delta x^2}{\Delta t} = c_s^2\left(\frac{1}{s_7}-\frac{1}{2}\right)\frac{\Delta x^2}{\Delta t} \qquad (4)$$

where c_s is the sound speed ratio, $1/\sqrt{3}$, and s_5 and s_7 are relaxation times.

3.2. Body Forces

The lattice Boltzmann equation in its original formulation does not account for body forces. Several approaches can account for body forces in LBM as summarized by Huang et al., 2011. We use the approach proposed by Guo et al., 2002, in which the body forces are added as an additional source term in the lattice Boltzmann equation

$$f_i(x+c_i\Delta t, t+\Delta t) - f_i(x,t) = [\Omega_i(x,t) + S_i(x,t)] \quad (5)$$

3.3. Turbulent Flow

Turbulence is accounted for using the Large Eddy Simulation (LES) approach. In this approach, scales smaller than the fluid mesh size are accounted for through an added subgrid scale eddy viscosity as a function of a subgrid scale tensor. We applied the wall-adapting local eddy-viscosity (WALE) model (Nicoud & Ducros, 1999) which is a function of the velocity gradient tensor.

The velocity gradient tensor can be calculated using second order finite differences—this requires only knowing the fluid velocity at the nearest neighboring fluid nodes. Using the WALE model, the subgrid scale viscosity is then calculated based on the velocity gradient at each node and added to the fluid viscosity at that node:

$$\nu + \nu_{sgs} = c_s^2\left(\frac{1}{s_5}-\frac{1}{2}\right)\frac{\Delta x^2}{\Delta t} = c_s^2\left(\frac{1}{s_7}-\frac{1}{2}\right)\frac{\Delta x^2}{\Delta t} \quad (7)$$

4. FLUID-SOLID COUPLING

There are several different approaches to describe boundary conditions for fluid-solid interaction. We adopted the so-called partially saturated method (PSM) of Noble & Torczynski, 1998, as modified by Holdych, 2003. As shown in Figure 3, as a solid particle moves through the fluid mesh it will partially or completely cover fluid cells—near the boundary of the solid particle, cells will be part solid and part fluid while they will be entirely solid in the interior of the particle. This is accounted for by modifying the lattice Boltzmann equation to have an additional collision operator for the solid phase:

$$\begin{aligned} &f_i(x+c_i\Delta t, t+\Delta t) - f_i(x,t) \\ &\qquad = (1-B)\Omega_i^f(x,t) + B\Omega_i^s(x,t) \qquad (8) \end{aligned}$$

where Ω_i^f is the fluid collision operator and Ω_i^s is a collision operator for solid nodes:

$$\Omega_i^s = \left(f_{\bar{i}}(x,t) - f_{\bar{i}}^{eq}(\rho,\vec{u})\right) - \left(f_i(x,t) - f_i^{eq}(\rho,\vec{u}_s)\right) \quad (9)$$

Here, $\vec{u}$ is the local velocity of the fluid and $\vec{u}_s$ is the velocity of the solid at point *x*; f_i^{eq} is the particle equilibrium population where the subscript $\bar{\imath}$ indicates the population with opposite velocity to *i.* The weighting factor, *B*, is determined based on the volumetric solid fraction, *ε*, of the fluid cell:

$$B(\varepsilon, s_5) = \frac{\varepsilon(s_5 - 1/2)}{(1-\varepsilon)+(s_5-1/2)} \qquad (10)$$

The additional collision operator accounts for the presence of the solid in the fluid, but the fluid action on the solid also needs to be accounted for. The hydrodynamic forces and torques on the solid are calculated according to Owen et al., 2010:

$$\vec{F}_p = \sum_n B_n \left(\sum_i \Omega_i^s \vec{c}_i \right) \qquad (11)$$

$$\vec{T}_p = \sum_n \left[\left(\vec{x}_n - \vec{X}_p \right) \times B_n \left(\sum_i \Omega_i^s c_i \right) \right] \qquad (12)$$

where x_n are all the lattice nodes covered or partially covered by the solid and X_p is the location of the center of mass of the solid particle.

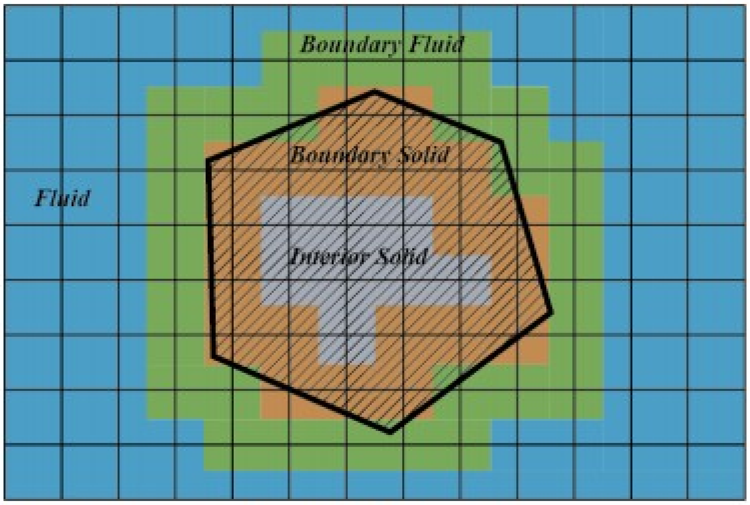


Figure 3: Node types for fluid-solid coupling

5. PARALLEL IMPLEMENTATION

Direct simulation of hydrodynamic forces on polyhedral blocks is computationally expensive, especially when simulations are three dimensional. Accurate resolution of the hydraulics surrounding the rock blocks requires the number of fluid nodes to be far greater than the number of solid particles. This makes the fluid domain solution the most computationally expensive portion of simulations and the most important part of the analyses to accelerate. As previously mentioned, the LBM is intrinsically parallelizable. The collision step is entirely localized for laminar flow and information only about a node’s nearest neighbors is required for turbulent flow. During the streaming step, information is shared only with nodes immediately surrounding each other—those nodes connected to each other by the discrete velocity set. This localized nature of the LBM is what makes it an attractive candidate for parallel computing.

Computing systems and hardware are evolving rapidly, providing tools to expand the types and sizes of analyses possible. However, these developments are heterogenous and do not necessarily interface easily with each other or maintain backward compatibility. When developing software that exploits a certain accelerator, it can be challenging and cumbersome to update the source code as the hardware that it targets is updated. To help alleviate some of the difficulties associated with this aspect of code development, the Kokkos C++ library (Edwards et al., 2014) abstracts both the data parallelism and memory access on a range of multi-core architectures. This allows the same source code to be compiled for different architectures while still maintaining performance.

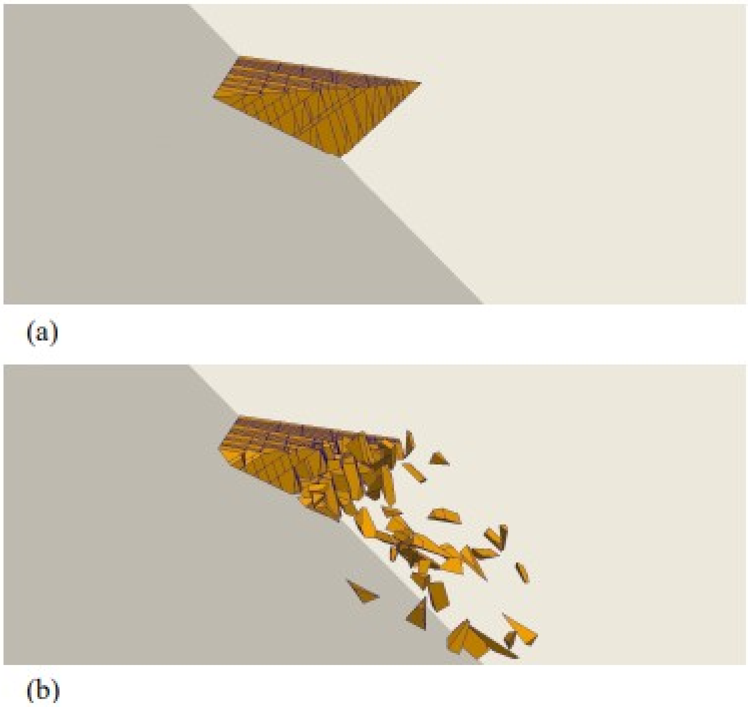
Using Kokkos, we parallelized the fluid portion of the computations such that the most computationally cumbersome calculations can be executed on either the central processing unit (CPU) or graphics processing unit (GPU). Approximately 30% of computing time is spent on the collision step, 30% on generating the output for visualizations and 20% on the fluid-solid coupling in terms of identifying which nodes are covered by solids moving through the fluid mesh. The collision step can be accelerated using either the CPU or GPU, while the fluidsolid coupling and output generation can be accelerated using the CPU.

6. EXAMPLES

The following examples are presented to demonstrate the capability of our software. It is capable of modeling coupled fluid-solid interaction but can also be used to model only fluid or dry rock. The figures in the examples were rendered using ParaView (Ayachit, 2015).

6.1. DEM Example: Rock Slope Failure

The source code for the application has been heavily unit tested to ensure that the underlying formulation is working as expected. Additionally, we have compared numerical results to the analytical solution for a block sliding down an inclined plane to verify the correctness of the DEM implementation. This provides confidence in the ability of the software to capture the behavior of more complex analyses, such as shown in the rock avalanche example in Figure 4. The initial configuration of the fractured rock mass was generated using SparkRocks (Gardner et al., 2017) using joint set data from a field site in the Sierra Nevada. The software is capable of capturing the behavior of the individual blocks and how they interact with each other as they move downslope.



(a)

(b)

Figure 4 Rock Avalanche: (a) Initial configuration; and (b) blocks sliding and rolling down slope.

6.2. LBM Example: Stagnation Point Analysis

The LBM portion of the source code has also been rigorously unit tested and compared to the analytical solution of several fluid dynamics problems—Couette flow, gravity-driven planar Poiseuille flow and flow down and inclined plane. Beyond that, we ran a simplified stagnation point analysis to verify that the software can correctly predict the location of the stagnation point compared to the work done by Frizell, 2007.

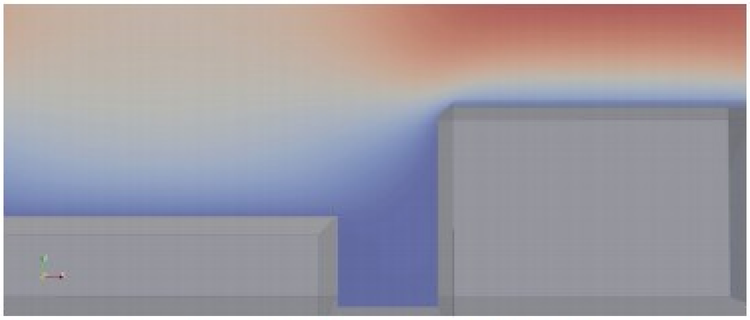


Figure 5: Stagnation Point Analysis—Velocity Magnitude

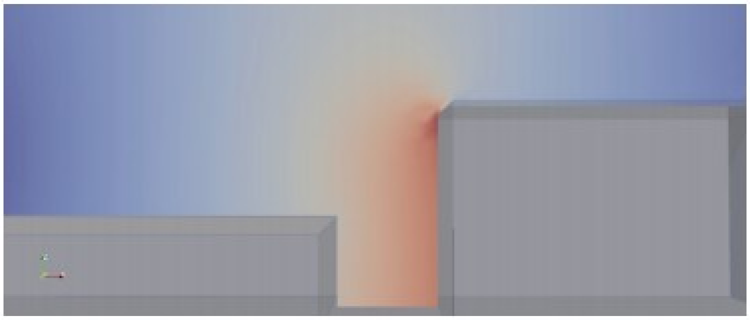


Figure 6: Stagnation Point Analysis—Pressure

Figure 5 shows the velocity magnitude in the channel and Figure 6 shows the accompanying pressure profile. Figure 7 shows stream tracers for a single snapshot in the solution—the stream tracers indicate how particles might travel through the velocity field at that instant in time. These results match the location of the stagnation point and show the recirculation zone identified in similar analyses by Frizell, 2007.

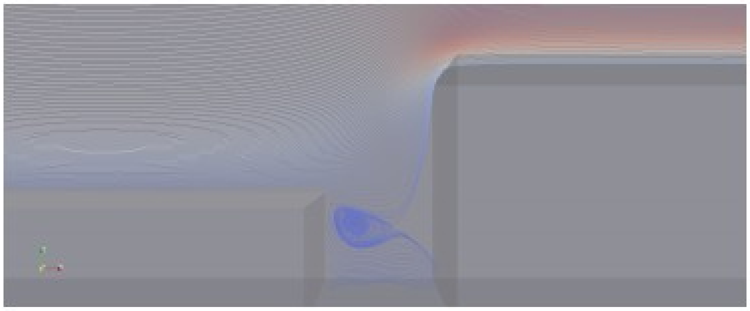


Figure 7: Stagnation Point Analysis—Stream Tracers

6.3. Coupled DEM-LBM Example: Block in Flow Down Inclined Plane

This example shows the results from a coupled, threedimensional analysis considering hydrodynamic forces on a polyhedral rock block in water flowing down n plane inclined at 30°. Figure 8 shows the block as it moves through the fluid mesh, effecting the fluid solution. Stream tracers show how fluid is forced to flow around the block and, in turn, applying hydrodynamic forces and moments to the block.

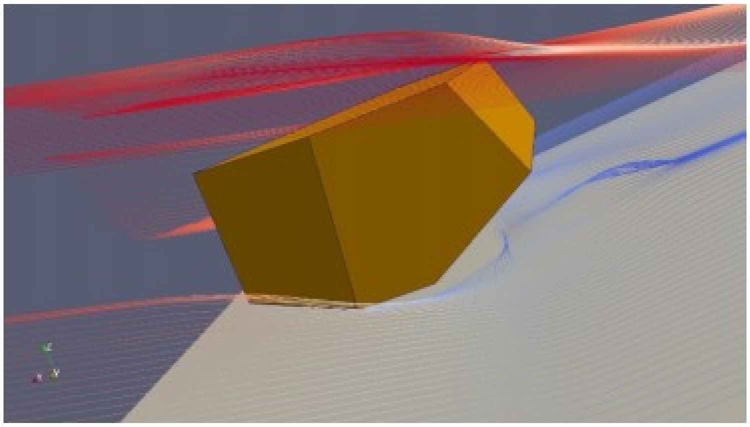


Figure 8: Block in water flowing down 30° inclined plane

7. CONCLUSION

We present the formulation for a coupled 3-D DEM-LBM solution of fluid-solid interaction capable of directly simulating hydrodynamic forces and moments acting on individual polyhedral blocks, along with preliminary results to demonstrate the capability of the methodology. The software implementation is structured so that it can be compiled to execute in parallel on either the CPU and GPU or CPU only. This is made possible by applying the Kokkos library to execute the most computationally expensive portions of the analyses.

Future work will be focused on further comparisons of coupled fluid-solid numerical analyses with analytical and experimental results to validate the accuracy of the numerical schemes. Ultimately, multiphase capability can be added to the LBM such that free-surface simulations involving rock falls and slides into water can be modeled as well. Additionally, the software will be expanded to have the capability to run on multiple compute nodes, whether on a super computer or on the Cloud, such that larger, real-world scale problems can be analyzed. This scalability will allow users to run large scale simulations on the Cloud, eliminating the need for investing and maintaining their own local computing resources.

References

1. Anderson, T.B. and R. Jackson. 1967. Fluid Mechanical Description of Fluidized Beds. I & EC Fundamentals. 6(4): 527–539. 2. Ayachit, U. 2015. The Paraview Guide: A Parallel Visualization Application. Kitware, ISBN 978-1930934306. 3. Bhatnagar, P., E. Gross and M. Krook. 1954. A Model for Collisional Processes in Gases I: Small Amplitude Processes in Charged and Neutral One-Component System. Phys. Rev. 94: 511. 4. Boon, C.W., G.T. Houlsby and S. Utili. 2012. A New Algorithm for Contact Detection between Convex Polygonal and Polyhedral Particles in the Discrete Element Method. Computers & Geotechnics. 44: 73–82. 5. Cundall, P.A. and O.D.L. Strack. A Discrete Numerical Model for Granular Assemblies. Géotechnique. 29(1): 47-

65. 6. Edwards H.C., C.R. Trott and D. Sunderland. 2014. Influence of Kinematics on Landslide Mobility and Failure Mode. J. Parallel. Dist. Comp. 74(12): 3202– 3216. 7. Frizell, K.W. 2007. Uplift and Crack Flow Resulting from High Velocity Discharges Over Open Offset Joints. USBR Report DSO-07-07. 8. Furuichi, M. and D. Nishiura. 2014. Robust coupled fluid-particle simulation scheme in Stokes-flow regime: Toward the geodynamic simulation including granular media. Geochemistry, Geophysics, Geosystems. 15(7): 2865– 2882. 9. Gardner, M.H., J. Kolb and N. Sitar. Parallel and Scalable Block System Generation. Computers and Geotechnics. 89: 168– 178. 10. Goodman, R.E. and G. Shi. 1985. Block Theory and its Application to Rock Engineering. London: PrenticeHall. 11. Guo, Z., C. Zheng and B. Shi. 2002. Discrete Lattice Effects on the Forcing Term in the Lattice Boltzmann Method. Phys. Rev. E. 65(4): 046308(6). 12. Hart, R., P.A. Cundall and J. Lemos. 1988. Formulation of a Three-Dimensional Distinct Element Method—Part II. Mechanical Calculations for Motion and Interaction of a System Composed of Many Polyhedral Blocks. Int. J. Rock Mech. Min. Sci. & Geomech. Abst. 25(3): 117– 125. 13. Holdych, D.J. 2003. Lattice Boltzmann Method for Diffuse and Mobile Interfaces. University of Illinois at Urbana-Champaign. 14. Hölzer, A. and M. Sommerfeld. 2009. Lattice Boltzmann Simulations to Determine Drag, Lift and Torque Acting on Non-Spherical Particles. Computers & Fluids. 38: 572– 589. 15. Horner, D.A., A. Carrillo, et al. 2000. Very Large Scale Coupled Discrete Element-Finite Element Modeling for Simulation Excavation Mechanics. Fourteenth Engineering Mechanics Conference, American Society of Civil Engineering, Austin, Texas. 16. Huang, H. M. Krafczyk and X. Lu. 2011. Forcing term in single-phase and Shan-Chen-type multiphase lattice Boltzmann models. Phys. Rev. E. 84(4): 046710(15). 17. Johnson, S.M., J.R. Williams and B.K. Cook. 2009. On the Application of Quaternion-based Approaches in Discrete Element Methods. Engineering Computations. 26(6): 610– 620. 18. Krüger, T., Kusumaatmaja, H., A. Kuzmin, O. Shardt, G. Silva, E.M. Viggen. 2016. The Lattice Boltzmann Method – Principles and Practice. Switzerland: Springer. 19. Munjiza, A. and K.R.F. Andrews. 1998. NBS Contact Detection Algorithm for Bodies of Similar Size. Int. J. Numer. Meth. Engng. 43: 131– 149. 20. Nicoud, F. and F. Ducros. 1999. Subgrid-Scale Stress Modelling Based on the Square of the Velocity Gradient Tensor. Flow, Turbulence and Combustion. 62: 183– 200. 21. Noble. D.R. and J.R. Torczynski. 1998. A LatticeBoltzmann Method for Partially Saturated Computational Cells. International Journal of Modern Physics. C. 9(8): 1189–1201. 22. Owen, D.R.J., C.R. Leonardi and Y.T. Feng. 2010. An Efficient Framework for Fluid-Structure Interaction using the Lattice Boltzmann Method and Immersed Moving Boundaries. Int. J. Numer. Meth. Engng. 87: 66– 95. 23. Shi, G. and R.E. Goodman. 1988. Discontinuous Deformation Analysis – A New Method for Computing Stress, Strain and Sliding of Block Systems. In Proceedings of the 29th U.S. Symposium on Rock Mechanics (USRMS), Minneapolis, Minnesota, 13-15 June 1988, eds. P.A. Cundall et al., 381–393. Rotterdam: Balkema. 24. Succi, S. 2001. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford: Oxford University Press.

25. Suga, K., Y. Kuwata, K. Takashima and R. Chikasue. 2015. A D3Q27 Multiple-Relaxation-Time Lattice Boltzmann Method for Turbulent Flows. Comp. & Math. Appl. 69: 518–529. 26. Tsuji, T., I. Akihito, and T. Toshitsugu. 2008. Multiscale Structure of Clustering Particles. Powder Technology. 179(3): 115–125. 27. Williams, J.R., E. Perkins and B. Cook. A Contact Algorithm for Partitioning N Arbitrary Sized Objects. Engineering Computations. 21 (2/3/4): 235–248.