Modeling of rock fracture flow using the Lattice Boltzmann Method on graphics hardware

S.A. Briggs & B.E. Sleep & B.W. Karney
University of Toronto, Toronto, Canada

ABSTRACT: Bioremediation has been accepted as a treatment technique for groundwater contamination in subsurface soils and shows promise for contaminated fractured rock environments. Biological growth in fractured rock is expected to occur predominantly as biofilms attach to the fracture surfaces. Biofilms in rock fractures are subject to a complex system of forces and other phenomenon due to the dynamics of the bulk fluid in which they grow. In this paper, through the applications of computational fluid dynamics (CFD) to rock fractures, where the boundaries are rough and the flow is complex, a precise analysis was conducted of the interaction of a fluid flow and the rock fracture. Specifically, hydraulic parameters and velocity profiles of an actual rock fracture were calculated and compared to a fracture of equivalent aperture. From the analysis it is clear that it is important to use more complex models such as the Lattice Boltzmann Method used is this paper to describe fracture flow.

1 INDRODUCTION

In this paper, flow through a single rock fracture is analysed through the development of a Lattice Boltzmann (LB) fluid model. Laboratory scale and in-situ testing is expensive and not always possible. Modeling the system allows the study of rock fractures and their role in bioremediation to be studied at a potentially faster pace. Although there is still a need for experimental evidence before any generalizations can be made, models allow the intelligent selection of potential experimental systems by trial and error on the computer, not in-situ.

In an effort to increase the model performance an alternate computer architecture was sought out. For several years, the graphics industry has been building massively parallel architectures for solving problems with large data sets. Today, the traditional graphics hardware is capable of performing general purpose computation and has begun to be harnessed by the scientific computing field. The newest Intel Core i7 processor is capable of 102.4 GFLOPS (1 billion FLOPS) while modern NVIDIA graphics hardware is cable of performing around 933 GFLOPS both with 32-bit accuracy. Graphics hardware achieves this performance through massive parallelization of hardware. For example some modern graphics hardware have 240 parallel processors and are able to feed these processors with sufficient memory bandwidth, 141.7 GB/s versus Intel’s 12.8 GB/s, to keep them busy. Now these Graphics Processing Units (GPUs) are capable of much more than running traditional video games, they are capable of general purpose scientific computing. NVIDIA has realized that it has a new customer in the field of scientific computing and has developed a new Software Development Kit (SDK) called CUDA (Compute Unified Device Architecture) which builds off the C programming language and runs on their G80 and newer generation of graphics hardware. CUDA enabled GPUs are relatively inexpensive and great for research on a budget. This paper presents a LB model implemented on such graphics hardware.

Lattice Boltzmann Methods (LBM) are discrete in time and space where many cells are used to allow the emergent behaviours of fluids to be observed. The discrete grids can easily be split up and distributed to many processors. Furthermore, it is the local nature of LBM that allows them to be effectively parallelized, since only next-neighbour communication is required, overhead is minimized. The GPU, which is also parallel in nature and is a perfect match to these types of models. Previous work has shown that an increase of an order of magnitude can be expected when implementing methods such as the LB approach on the GPU (Tolke 2008, Bailey et al. 2009).

This paper presents a two dimensional LBM code implemented on the GPU. It is the intention to move forward in the next paper with the inclusion of a biofilm model using cellular automata (CA). This will create a biofilm-fluid interaction model. Therefore consideration to the boundary between the two models is of utmost importance. Since both models have roots in CA, and are both discrete in nature, the grids of both models will have to be the same scale to simplify calculations between models. During a simulation as heterogeneous colonies of bacteria grow outwards from the surface they are expected to extend past the viscous sub-layer where they will be subject to shearing and pressure gradients simulated by the LB model. Likewise, the fluid will have to move around the biofilm, reducing the effective hydraulic aperture and causing recirculation. These dynamic fluid phenomena could cause biofilm streamers as seen experimentally by Stoodley et al. (1998). Detachment mechanisms have also been theorized to occur due to shear stresses applied to the biofilm by the fluid. CFD models such as the LB method developed here would be able to determine shear distributions through careful treatment of velocity and density profiles.

2 BACKGROUND

Lattice Boltzmann Methods are types of numerical methods for solving CFD problems. Other types of CFD start with the Navier-Stokes equations, which govern the macroscopic movement of fluids, then discretize to get a solution to a system of PDEs (Eker & Akin 2006). In the LBM model the microscopic interaction of particles on a grid and the averaging of those interactions emerge into the macroscopic continuum of a fluid. These interactions include two main steps: streaming and collision. The streaming step is a translation of particles from one node on the grid to the next. The collision step conserves momentum by redirection of particles which ‘collide’ or occupy the same node.

Originally, when CA methods applied to fluid dynamics, the Lattice Gas Automata (LGA) methods were developed. LGA methods are discrete in space, time and particle velocity. The first hexagonal grid, with 7 particle velocities, was developed by Frisch et al. in 1986. This model consisted of a lattice for which each node at 6 vertices connected to other nodes. The seventh particle velocity came from the stationary case, or zero velocity. In Frisch’s model, there could be either 0 or 1 particle at any given node moving, or streaming, in any direction. The collision step would occur when more than one particle would occupy the same node. The collision rules would conserver mass, momentum and energy before and after each collision.

LBM evolved from LGA to address some of its short-comings, the primary one being the Boolean treatment of particles at a node. Instead, LB methods use a particle distribution function to describe the nodal velocities (Martys 2007). In two dimensions, nine velocity directions e_i where i = 0,1,2...8 are sufficient to describe a continuum fluid. Each node has 8 vertices and e_o represents a particle at rest. The naming convention used for LBM is D*d*Q*q*, where *d* represents the dimension and *q* represents the velocities (Sukop & Thorne 2005). In this case the model would be D2Q9 for a two dimensional lattice using nine velocities. The velocities vary such that each particle may travel one lattice unit (*lu*) each time step (*ts*).

The velocity distribution function, *f*, represents the frequency of a particle occurring in any of the nice discrete velocities. The frequencies correspond to the density of fluid in any given direction. Therefore one can derive the macroscopic fluid density to be the sum of all the velocity distribution functions (Sukop & Thorne 2005) as shown in Equation 1.

$$\rho = \sum_{i=0}^{8} f_i \tag{1}$$

Similarly, Equation 2 shows the macroscopic velocity **u** is an average of all the discrete velocities weighted by the velocity distribution function, *f*, or their probability densities (Sukop & Thorne 2005).

$$\mathbf{u} = \frac{1}{\rho} \sum_{i=0}^{8} f_i e_i \tag{2}$$

Using Equation 1 and 2 the microscopic quantities can be related to the desired macroscopic velocity. The streaming and collision step now has to be considered with more complexity than was done with the LGA method. The streaming is done in a similar method, a translation of particles, however the collision rules are replaced with a continuous function. A popular collision function is the Bhatnagar-Gross-Krook (BGK) relaxation term in Equation 3 (Sukop & Thorne 2005). The velocity distribution function tends to the equilibrium distribution according to the BGK collision term (Wagner 2005).

$$\Omega_i = -\frac{1}{\tau}(f - f_i^{eq}) \tag{3}$$

where f_i^{eq} is the local equilibrium value for the velocity distribution function in the direction of link e_i and varies depending on the lattice used. In the BGK model, the fluid tends towards equilibrium at a rate governed by the relaxation term, τ (Latt 2008a). The BGK collision operator expressed above along with the streaming step, which is a discretization of the Boltzmann equation, is one of the simplest forms of the a LBM and is given in Equation 4:

$$f_i(x + e_i\Delta t, t + \Delta t) - f_i(x,t) = \Omega_i \tag{4}$$

where *x* represents the position and *t* represents time. The function $f_i(x,t)$ is the original distribution function at time *t* and $f_i(x+e_i\Delta t,t+\Delta t)$ is the distribution function at time $t+\Delta t$. Over that time, a LB particle has moved a distance of $e_i\Delta t$ or to the next node in the direction of e_i (Brewster 2007). The equilibrium distribution function used in the BGK collision term is described by Eqution 5.

$$f_i^{eq}(x) = w_i\rho(x)\left[1 + 3\frac{e_i \cdot \mathbf{u}}{c^2} + \frac{9(e_i \cdot \mathbf{u})^2}{c^4} - \frac{3\mathbf{u}^2}{2c^2}\right] \tag{5}$$

where w_i are weights (4/9 for i=0, 1/9 for i =1,2,3,4 and 1/36 for i =5,6,7,8) and c is the lattice speed which has a value of unity for simple implementations (Sukop & Thorne 2005).

LBM are essentially explicit finite difference approximations of the Boltzmann equation and using a Chapman-Enskog expansion, the Navier-Stokes equations for incompressible flow can be recovered (Eker & Akin 2006). The LBM are typically 1st order accurate in time and 2nd order accurate in space depending on the implementation of the collision term (Tolke 2008).

LBM methods, which originate from a CA structure, are efficiently parallelized in computer programming due to the locality of the discretization. Each node is only concerned with its direct neighbours and therefore when the lattice is distributed to parallel processors the only required communication is at the sub-lattice boundaries (Martys 2002).

3 MODEL IMPLEMENTATION

3.1 *Lattice Boltzmann Method*

The model created is a 2D LBM using a BGK collision operator as previously discussed and summarised in Equation 6:

$$f_i(x + e_i\Delta t, t + \Delta t) - f_i(x,t) = -\frac{1}{\tau}(f(x,t) - f_i^{eq}(x,t)) \tag{6}$$

where the left hand side of the equation represents the streaming step and the right hand side represents the collision step.

The model runs the LBM as a general purpose GPU (GPGPU) program through the CUDA API. Real-time visualization is also run on the GPU using OpenGL. The model was based on similar techniques found in Latt (2008a), Sukop & Thorne (2005) and Tolke (2008).

One of the drawbacks of GPU implementations is the limitation to 32 bit floating point precision. Current hardware has limited support for double or 64 bit precision calculations and when used, most of the performance benefit of using the GPU is lost. Without double precision calculations, numerical dissipation is likely occurring in the model being presented. Numerical dissipation is the artificial dissipation of momentum in the fluid due to numerical error. Since the LBM is essentially a finite difference approximation to the Boltzmann equation, it is subject to the same numerical truncations as other finite difference methods. The numerical error can cause dissipation of the advection term which by definition should be free of dissipation (Zhu et al. 2006). The advection term in the LBM is represented by the streaming step or uniform translation of data. Since the convection term is also treated in the same streaming step by LB methods (Yu 2003), the LBM model presented along with other LBM models can run into numerical difficulties.

To minimize the potential for numerical instabilities and maintain the second order accuracy of the LB method, the model parameters were defined using the method laid out by Jonas Latt as part of the OpenLB User Guide (2008). The process involves selecting physical units then converting to lattice units to finally obtain the relaxation parameter τ. The relaxation parameter plays an important role in the collision term of the LBM. It controls the tendency of the system to move towards local equilibrium. In the literature (Sukop & Thorne 2005), the relaxation parameter has been found to cause numerical instabilities at values approaching 0.5 from the right hand side (τ must be greater than 0.5 for physical viscosities). The LB model presented is slightly more sensitive needing a value closer to 0.6 for stability. Stable values of τ close to unity are preferred for simple implementation of the LBM (Skukop and Thorne 2005) and can be found using the method outlined below.

In this research water is the physical fluid being simulated with a kinematic viscosity, ν in a fracture of width h with physical velocity u. This leads to an expression for the Reynolds number in Equation 7.

$$R_e = \frac{2h \cdot u}{\nu} \tag{7}$$

The dimensionless expression for Reynolds number is then used to convert from the physical units of the system to lattice units. The fracture width is discretized into lattice nodes of length δ_x. To minimize the slightly compressible nature of the LB methods the constraint in Equation 8 is used to establish δ_t. Also, in order to maintain second order accuracy, it can be shown that the relationship between δ_x and δ_t must follow Equation 9 (Latt 2008b). The constraints on time discretization are shown below:

$$\delta_t < \frac{\delta_x}{\sqrt{3}} \tag{8}$$

$$\delta_t \sim \delta_x^2 \tag{9}$$

The lattice viscosity (ν_L), is calculated based on the discretization of the system and the dimensionless Reynolds number. Finally, the relaxation parameter is calculated according to Equation 10 and is kept as close to unity as possible as mentioned previously.

$$\tau = 3 \cdot \nu_L + 1/2 \tag{10}$$

3.2 *Boundary Conditions*

One of the distinct advantages of the LBM comes from its discrete nature. It is efficient for modeling complex geometries (Chen et al. 1994, Eker et al. 2006, Lammers et al. 2006, Brewster 2007) which arises in the analysis of rock fractures. A Boolean array is stored to set the value of any point in the LB grid to represent either a fluid cell or a solid boundary. At the solid boundaries, a no-slip condition is used to create zero velocity at the boundary surface. A different set of collision equations are used as described by Succi (2001) and are referred to as mid-plane bounceback boundary conditions. The name arises from the applied boundary rules. Particles entering a boundary at time t are sent back out with equal magnitude and opposite direction at time $t+\Delta t$ this effectively puts the boundary at a distance midway between a fluid and solid node.

Two types of input boundary conditions are used, both being of Von Neumann type. These boundaries set the velocity and density of the first column of nodes within the model, creating a constant influx of fluid. The first input boundary is a constant velocity boundary where all nodes in the first column are set to a desired velocity. This is similar to what occurs at the beginnings of actual rock fractures or in general flow between parallel plates. Secondly, a Poiseuille input velocity profile can also be used. This describes flow far downstream of any inlet or fully developed flow between parallel plates. The outlet boundary condition is a zero gradient condition, which essentially allows the fluid to flow out of the model without effecting the upstream flow as long as the outlet is sufficiently downstream of the inlet.

4 RESULTS

4.1 *Flow Between Parallel Plates*

Before any interesting observations can be made using the LBM model, it is important to first ensure that it is correctly modeling incompressible flow. By comparing the model with analytical solutions of known flow conditions, some validation of the model can be done. The first flow condition is that of flow between parallel plates. For laminar flow, the Hagen-Poiseuille equation can be used to describe the horizontal velocity through a cross-section. This analytical solution yields a parabolic velocity profile. The model also creates a parabolic velocity profile, however the amplitude of the profile does vary some as shown in Figure 1. Figure 1 shows a selection of horizontal velocity profiles plotting the ratio of velocity (u) to maximum velocity (U) at all nodes across the model domain. The domain is 256 nodes wide by 1024 nodes in length (L). The L/4 profile is almost exactly the same as the Poisseuille profile while the 3L/4 profile shows a 2% reduction in maximum velocity along the centre line of the system. This could be due to numerical dissipation from the use of single precision calculations.

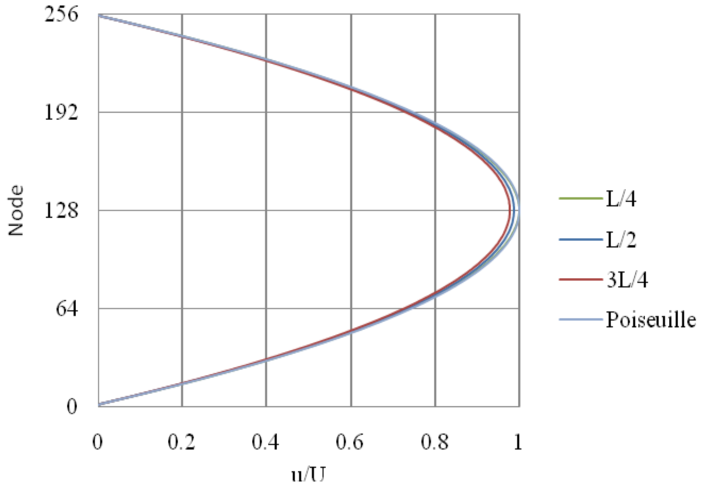

Figure 1. Horizontal velocity profile through a selection of cross-sections developed from a Poiseuille inflow boundary condition.

Figure 1 was developed using a Poiseuille input velocity profile, but it is also interesting to see how long it takes the model to develop the same profile on its own. In Figure 2 the development of the parabolic profile using a constant input velocity is plotted. In this case L is 2048 and only 128 vertical nodes are used. It can be seen that it takes longer to fully develop the flow, which is to be expected. At a distance of 3L/4 or 12 widths the Poiseuille velocity is reproduced. Similarly to Figure 1, it can be seen that the further downstream the higher the relative error, and in this case, at the end of the model domain, a relative error of 4.6% is calculated. This is larger than that found in Figure 1 but could be due to the longer domain used for Figure 2 leading to further numerical dissipation.

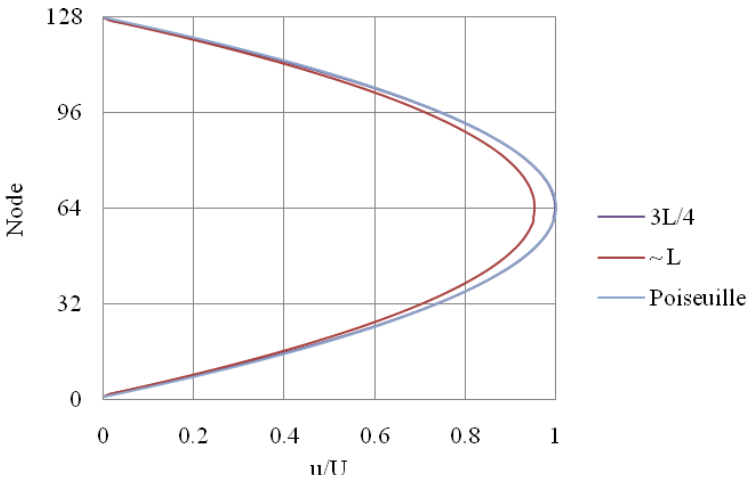

Figure 2. Horizontal velocity profile through a selection of cross-sections developed from a constant inflow boundary condition.

4.2 *Flow Over a Backward Facing Step*

The next case for comparison is that of flow over a backward facing step. Typical geometry for a backward facing step flow case uses a step height (*S)* that is half the total height of the system, and a length at least 30 times the height which ensures that the outflow boundary condition does not affect the flow geometry. The inlet boundary condition is a constant parabolic velocity profile and the outlet is a zero-gradient boundary. All other boundaries use standard bounce-back rules. This case is well studied but typically at higher Reynolds numbers than is necessary for the study of flow in fractures. Therefore only the results presented by Armaly et al. (1983) for *Re ≤ 200* are used. The relationship of the dimensionless Reynolds number is described in

Equation 7 where *2h* characteristic length for flow between parallel plates, the upstream height or step height is used. Next, *u* is the average inlet velocity and finally, ν is the kinematic viscosity of water.

Qualitatively, Flow over a backward facing step at Reynolds numbers under 200 consists of an area directly after the step of recirculation followed by a reattachment point and finally development of a parabolic velocity profile downstream of the step. The reattachment point refers to the end of the recirculation zone. Figure 3 also shows this qualitatively, the flow in Figure 3 is from left to right, blue represents the slowest velocities while red the fastest. The inlet boundary condition is parabolic and the outlet is a constant gradient. The step height is half of the total height and the Reynolds number is 100. It can be seen that the reattachment length is approximately equal to that reported by Armaly et al. (1983) of 3 times the step height and the velocity profile becomes parabolic again far downstream.

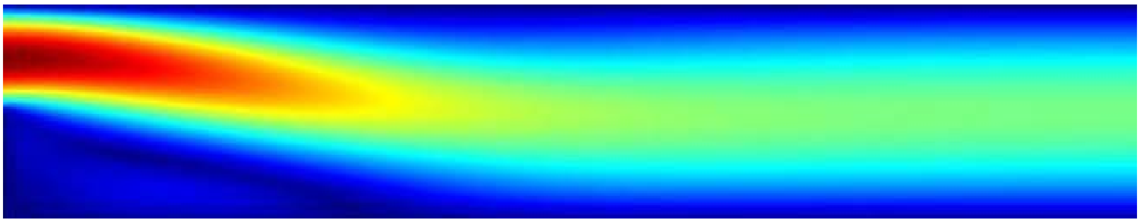

Figure 3. Flow over a backward facing step, Re=100.

At higher Reynolds numbers, the model does initially predict the reattachment length as found by Armaly et al. (1983) however the simulations quickly become unstable. The instability is due in part to the relaxation parameter in the LBM model which approaches 0.5 for both *Re=150* and *Re=200*, at which point the LBM model falls apart numerically. This is typical of LBM BGK models (Sukop & Throne 2005). Adjustments can be made to the discretization of the grid to compensate for the change in relaxation parameter, but it is at the cost of the accuracy of the model. Furthermore, it appears that the constant gradient outflow boundary may play a role in shaping the upstream flow also found by Succi (2001). During simulation, a distinct shockwave moves backwards from the outflow boundary and changes the geometry of the flow over the step. Finally, numerical dissipation due to the use of single precision calculations could be the cause of additional errors.

For the purposes of subsurface water flow in rock fractures Reynolds numbers are typically less than one, both model stability and accuracy can be maintained. Should the model be required to scale to higher velocities, care would have to be taken with the treatment of the LBM model including the use of double precision calculations and improved boundary conditions. At low Reynolds numbers, less than one, recirculation zones still occur after a backward facing step. This shows that although the flow is laminar, it is not necessarily simple and therefore treatment of the flow by an advanced model such as the LBM is appropriate.

4.3 *Flow Through Rock Fractures*

After applying the model to analytical test cases and knowing it performs well for the range of Reynolds numbers required, we can extend the model to simulate flows of more complex systems. Rock fractures offer complex boundaries which are well suited to the LBM model being used. Fracture data can be input into the model to simply create solid nodes wherever the fracture is found. Data from one side of a fracture that was collected by Boutt et al. (2006) is used in the LBM model. The data consists of aperture values along a 2D slice through an actual fracture, it is measured from a reference datum which was a smooth plate in the case of Boutt et al. (2006). Of interest in this paper is the equivalent aperture of a given rock fracture which is used to approximate the complexity of the fracture into one term so that it may be used to calculate the flow through, or the discharge velocity of a fracture. The Cubic Law was chosen to compare against the LBM model. The Cubic Law corresponds to the geometric mean of a given fracture aperture data set and is accurate for the purposes of hydraulic calculations (Zheng et al. 2008).

For comparison, the LBM model was run first as flow through the fracture and then as flow through parallel plates with a separation equal to the equivalent aperture calculated using the geometric mean of the fracture aperture data set. Table 1 summarizes the fracture parameters, discharge velocities and flow predicted by the equivalent fracture and that found from the actual fracture data, both using the LBM model. The inlet boundary condition was fully developed parabolic flow with an average velocity of 0.0002 m/s. As expected, the geometric mean accurately describes hydraulic parameters such as flow rate through a fracture to within 3% in the presented model.

Table 1. Comparison of the Geometric mean and actual fracture data using LBM model.

	Geometric Mean	Actual Fracture	Relative Error
Overall Aperture: m	3.59 x10^{-4}	2.94 - 4.76 x10^{-4}	--
Overall Aperture: Nodes	193	158 - 256	--
Outlet Aperture: m	3.59 x10^{-4}	4.76 x10^{-4}	--
u_{max} at outlet: m/s	2.92 x10^{-4}	3.01 x10^{-4}	3.0%
u_{avg} at outlet: m/s	2.00 x10^{-4}	1.50 x10^{-4}	25%
Flow: m^3/s	1.31 x10^{-13}	1.35 x10^{-13}	3.0%

More importantly, however, when modeling biofilms or other complex phenomena within a rock fracture, it is what occurs inside the model that is interesting, not simply the average flow characteristics. The model offers real insight into the dynamics within the rock fracture, whereas the Cubic Law cannot. Figure 4 plots a selection of velocity profiles through the equivalent fracture as well as the actual fracture. Although the final velocities may be equivalent as seen in Table 1, at various locations through the fracture the velocity profiles differ significantly. The future research this paper is working towards requires more fine grained analysis of these velocities throughout the system in order to better quantify the effect on biofilms that may be present in the fracture.

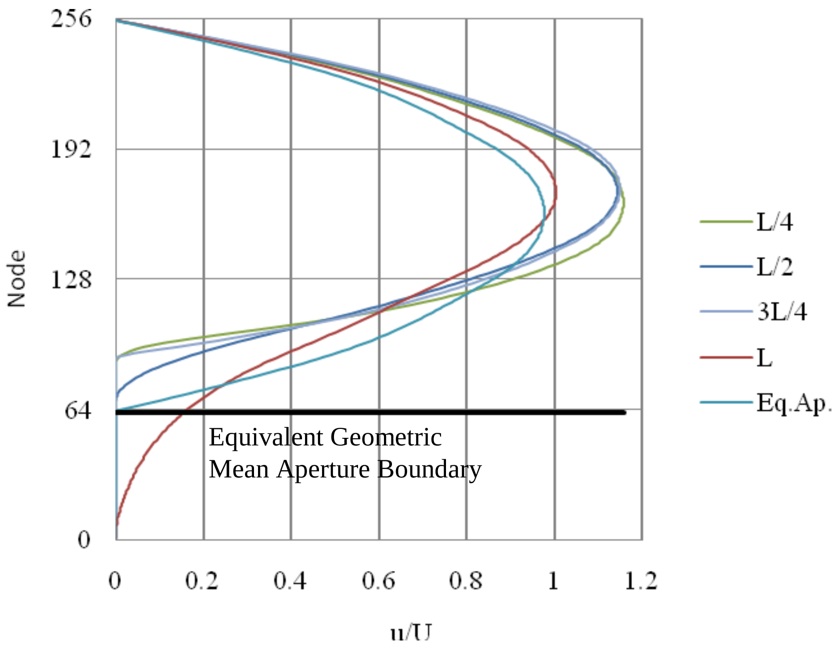

Figure 4. A selection of velocity profiles of the actual aperture compared to the equivalent geometric mean aperture boundary velocity profile.

Figure 5 compares the results from the two separate simulations discussed above. The left hand side of Figure 5 consists of a rock fracture along the base of the model with a no-slip smooth top boundary, constant gradient outlet and parabolic inlet boundaries. The right hand side of Figure 5 models flow through parallel plates spaced at an equivalent aperture calculated using geometric mean of the fracture data. It can be seen that the actual rock fracture compresses the velocity profile much more than that of the equivalent fracture. It is the peaks of the rock fracture that significantly change the velocity distribution, leading to an apparently smaller equivalent aperture than that found by the geometric mean which corresponds to the Cubic Law aperture. The flow distribution is clearly different from that predicted by simple parallel plates and although it cannot be seen in Figure 5, there are areas of recirculation downstream of each fracture constriction. How this would affect a biofilm or perhaps nutrient concentration is poorly understood and the subject of future research. Since this is a complex phenomenon, it would be difficult to create a single variable that could adjust for such effects. Rather, it is important that a given system be simulated with a model of equal complexity such as the presented LBM model.

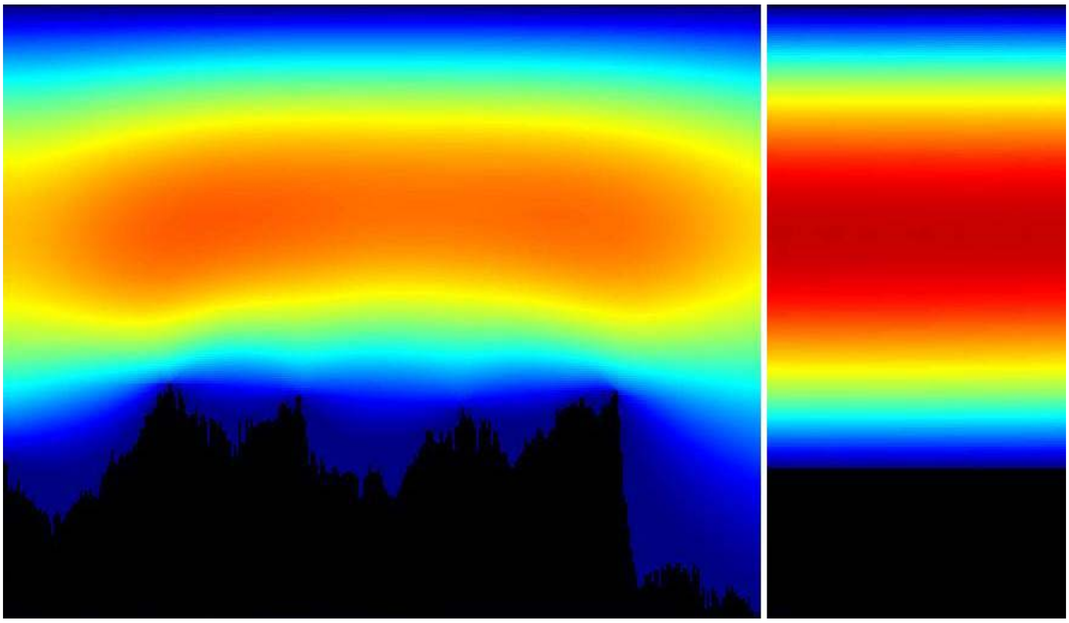

Figure 5. On the left: Flow through a fracture. On the right: Flow through parallel plates with the geometric mean aperture equivalent to the actual aperture on the left.

5 DISCUSSIONS & CONCLUSIONS

Performance of the LBM on the GPU is much faster, roughly an order of magnitude, than a comparable LB model running on a CPU, consistent with the findings of Tolke (2008a). It is the hope of this paper to lend insight into the various computer architectures that are available to engineers for high performance computing. It is possible for any researcher to now harness tremendous computing power. While the GPU is used for general computation in this research it is also used for real-time visualization. The model developed for this paper is well suited for simulating laminar flows through simple systems like parallel plates, and more complex system such as rock fractures. It can model internal flow dynamics that are lost to other types of flow approximation like the Cubic Law because it takes into account the complex boundaries that arise in rock fractures. Even in laminar flow, recirculation occurs, creating potentially interesting phenomenon for the interaction of biofilms in those areas leading to exiting future research into this area.

REFERENCES

Armaly, B.F. & Durst, F. & Pereira, J.C.F. & Schonung, B. 1983. Experimental and theoretical investigation of backward-facing step flow. *J. fluid mech.* 127: 473-496.

Bailey, P. & Myre, J. & Walsh, S.D.C. & Lilja, D.J. & Saar, M.O. 2009. Accelerating Lattice Boltzmann fluid flow simulations using graphics processors. *Submitted to the second work shop on general-purpose computation on graphics processing units (GPGPU2), ACM Inter national conference on architectural support for programming languages and operating sys tems (ASPLOS).*

Brewster, J.D. 2007. Lattice Boltzmann simulations of three-dimensional fluid flow on a desktop computer. *Analytical Chemistry*. 79(7): 2965-2971.

Boutt, D.F. & Grasselli, G. & Fredrich, J.T. & Cook, B.K. & Williams, J.R. 2006. Trapping zones: the effect of fracture roughness on the directional anisotropy of fluid flow and colloid transport in a single fracture. *Geophysical Research Letters.* 33.

Chen, S. & Doolen, G.D. & Eggert, K.G. 1994. Lattice Boltzmann fluid dynamics: a versatile tool for multiphase and other complicated flows. *Los Alamos science*. 22: 98-111.

Eker, E. & Akin, S. 2006. Lattice Boltzmann simulations of fluid flow in Synthetic fractures. *Transport in porous media*. 65: 363-384.

Frisch, U. & Hasslacher, B. & Pomeau, Y. 1986. Lattice-gas automata for the Navier-Stokes equation. *Physical review letters*. 56(14): 1505-1508.

Lammers, P. & Beronov, K.N., Volkert, R. & Brenner, G. & Durst, F. 2006. Lattice BGK direct numerical simulation of fully developed turbulence in incompressible plane channelflow. *Computers & fluids*. 35: 1137-1153.

Latt, J. 2008a. *OpenLB user guide associated to release 0.4 of the code*. Tufts: Tufts Univer sity.

Latt, J. 2008b. *Choice of units in Lattice Boltzmann simulations*. Tufts: Tufts University.

Martys, N.S. & Hagedorn, J.G. 2002. Multiscale modeling of fluid transport in heterogeneous materials using discrete Boltzmann methods. *Materials and structures.* 35:650-659.

Stoodley, P. & Lewandowski, Z. & Boyle, J.D. & Lappin-Scott, H.M. 1998. Oscillation charac teristics of biofilm streamers in turbulent flowing water as related to drag and pressure drop. *Biotechnology and bioengineering*. 57(5): 536-544.

Succi, S. 2001. *The Lattice Boltzmann equation for fluid dynamics and beyond.* Oxford: Oxford University Press.

Sukop, M.C. & Thorne Jr., D.T. (2nd) 2005. *Lattice Boltzmann modeling: An introduction for geoscientists and engineers*. Berlin: Springer.

Tolke, J. 2008. Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture Developed by nVIDIA. *Computing and visual sci*. *Preprint submitted to Springer*.

Tolke, J. & Krafczyk, M. 2008. TeraFLOP computing on a desktop PC with GPUs for 3D CFD. *International journal of computational fluid dynamics.* 22(7): 443-456.

Wagner, A. 2005. *A practical introduction to the Lattice Boltzmann method*. Fargo: North Da kota State University.

Yu, D. & Mei, R. & Luo, L. & Shyy, W. 2003. Viscous flow computations with the method of Lattice Boltzmann equation. *Progress in aerospace sciences*. 29: 329-367.

Zheng, Q. & Dickson, S.E. & Guo, Y. 2008. On the appropriate "equivalent fracture" for the description of solute transport in single fractures: laboratory-scale experiments. *Water resources research*. 44.

Zhu, H. & Xuehui, L. & Liu, Y. & Wu, E. 2006. Simulation of miscible binary mixtures based on Lattice Boltzmann method. *Computer animation and virtual worlds*. 17: 403-410.