Flow Through Porous Media

Arjun Chintapalli

Final Project for Computational Fluid Mechanics ME 7751

at Georgia Institute of Technology

5/3/2017

Table of contents

I. ABSTRACT

Matlab code was implemented to create Lattice Boltzmann flow simulations of porous media in 2D channel flow. The Lattice Boltzmann method was implemented utilizing a D2Q9 lattice and Bhatnagar–Gross–Krook collision operator under both periodic boundary conditions as well as Zou-He boundary conditions to simulate Pouseuille flow. No-slip boundary conditions as well as immersed boundary conditions were implemented through the use of bounce-back boundary conditions. A body force was utilized to drive the fluid flow. Results were achieved for cases of flow around cylinder, flow through porous media and flow through a sinusoidal pore.

II. INTRODUCTION

Having a Petroleum Engineering background, I was extremely excited to use the Lattice Boltzmann Method to simulate fluid flow through porous rock media. Fluid simulations are extremely important in this field to help determine the ultimate recovery of hydrocarbons from oil reservoirs under various scenarios and parameters. To implement this objective, I chose to implement my fluid simulator in the following steps:

1. Poiseuille Channel Flow
2. Immersed Boundary Conditions to Simulate Fluid Obstacles
3. Porous Media Validation

III. LITERATURE REVIEW

A review of literature revealed that there is extensive application of the Lattice Boltzmann Method to model oil, gas and water flow through porous media.

Nearly all of the literature broke down their implementation steps as follows:

1. Poiseuille Channel Flow
2. Immersed Boundary Conditions to Simulate Fluid Obstacles
3. Multiphase Implementation
4. 3D implementation

Optionally, multiple relaxation time, parallelization, and varied fracture geometry were also found to be implemented. General applications in the oil and gas industry of the Lattice Boltzmann Method were to model residual saturation profiles, relative permeability curves and pressure required to move an oil slug. Although I attempted to do the first three steps, I could successfully only implement the first two steps as outlined in this report.

IV. COMPUTATIONAL MODEL

The Lattice Boltzmann Method relies on modelling random but biased molecular interactions by analysing probability distribution functions at nodes to simulate fluid flow. The setting up of the Lattice Boltzmann Method can be broken down into the

three steps of equation setup, lattice pattern construction and boundary node definition.

4.4 LATTICE BOLTZMANN EQUATION SETUP

With the appropriate choice of collision operator, the Lattice Boltzmann Method method will recover the Navier-Stokes equation used aprior in this course. The Navier-Stokes transport equations used so far are:

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} + \nabla.(uu) + \nabla(P) - v\nabla^2(u) = 0$$

These equivalent Boltzmann equations are:

$$\frac{\partial f}{\partial t} = (\frac{\partial f}{\partial t})_{Force} + (\frac{\partial f}{\partial t})_{Diffusion} + (\frac{\partial f}{\partial t})_{Collision}$$

$$\frac{\partial f}{\partial t} + u.\nabla(f) = (\frac{\partial f}{\partial t})_{Collision}$$

$$\mathrm{f_\alpha}(\vec{\mathrm{x}} + \mathrm{e_\alpha}\Delta\mathrm{t}, \mathrm{t} + \Delta\mathrm{t}) = \mathrm{f'_\alpha}(\vec{\mathrm{x}}, \mathrm{t}) + \frac{\Delta\mathrm{t}}{\mathrm{N_\alpha e^2}} \mathrm{e_{\alpha i}}\, \mathrm{F_i}(\vec{\mathrm{x}}, \mathrm{t})$$

$$\mathrm{f_i}(\vec{\mathrm{x}} + \mathrm{e_i}\Delta\mathrm{t}, \mathrm{t} + \Delta\mathrm{t}) - \mathrm{f_i}(\vec{\mathrm{x}}, \mathrm{t}) = -\frac{\mathrm{f_i}(\vec{\mathrm{x}}, \mathrm{t}) - \mathrm{f_i}^{eq}(\vec{\mathrm{x}}, \mathrm{t})}{\tau}$$

This above equation breaks down the Lattice Boltzmann calculation to two steps for the inner nodes: the propagation step on the left and the collision step on the right, the the $\mathrm{f_i}^{eq}$ term represents the equilibrium distribution and the τ term represents the relaxation time. This discretization implies an explicit scheme that is second order in

space and first order in time. The Navier-Stokes Equation mentioned prior can be directly recovered from this equation through Chapman-Enskog expansion as discussed in class. The above equation simplifies to this

$$f_i(\vec{x} + \vec{c}_i, t + 1) = \frac{f_i(\vec{x}, t)(\tau - 1) + f_i^{eq}(\vec{x}, t)}{\tau}$$

Then, the macroscopic ρ, and $\vec{u}$ are calculated as follows:

$$\rho(\vec{x}, t) = \sum_{i=0}^{8} f_i(\vec{x}, t)$$

$$\vec{u}(\vec{x}, t) = \frac{1}{\rho}\sum_{i=0}^{8}(c * f_i(\vec{x}, t) * e_i)$$

For this project, the Bhatnagar-Gross-Krook collision operator was chosen, and thus the f_i^{eq} term simplifies as follows:

$$f_i^{eq}(\vec{x}, t) = w_i p\left(1 + \frac{3e_i \cdot u}{c} + \frac{9(e_i \cdot u)^2}{2c^2} - \frac{3(u)^2}{2c^2}\right)$$

The weighting parameter, w_i , varies with direction, i , such that:

$$w_i = \begin{cases} \frac{4}{9} & i = 0 \\ \frac{1}{9} & i = 1,2,3,4 \ [Cardinal] \\ \frac{1}{36} & i = 5,6,7,8 \ [Ordinal] \end{cases}$$

Additionally, c is the lattice speed and is calculated such that:

$$c = \frac{\Delta x}{\Delta t}$$

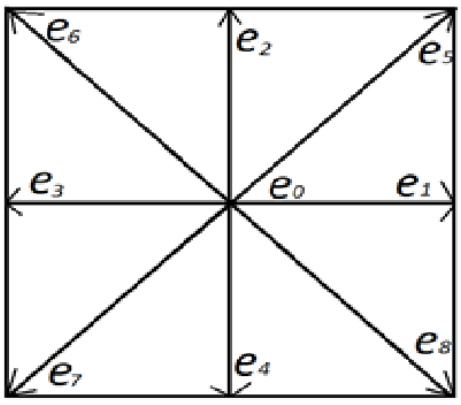
Since Δt is one, the lattice speed varies with the size of the mesh.

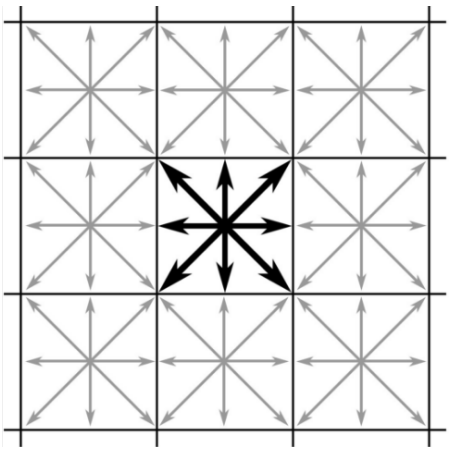
τ, the relaxation parameter, is calculated as follows:

$$\tau = \frac{1}{2}(6 * \frac{u}{Re}\frac{\Delta t}{\Delta x^2} + 1)$$

4.5 LATTICE BOLTZMANN PATTERN SETUP

The Lattice Boltzmann Method involves confining particles to nodes within a lattice pattern and then simulating particle movements across the nodes in the lattice. For this homework, a D2Q9 lattice pattern setup was implemented, whereby the lattice is in two-dimensional framework and each node has 9 velocity directions. The following illustration shows a lattice node with its component velocity vectors on the left and the wider lattice framework on the right.





4.6 LATTICE BOLTZMANN BOUNDARY CONDITION SETUP

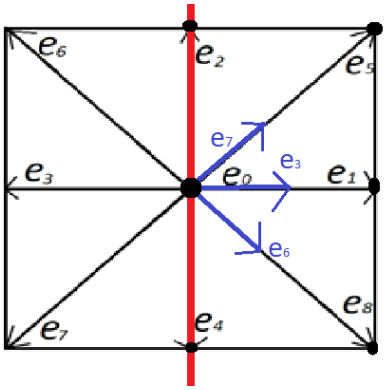
A simple bounce back boundary condition was implemented on the walls and also on the immersed boundaries. The immersed boundaries were created by creating a Boolean array the same size of the grid, and then setting the nodes boundaries to be 1

and the fluid region to be 0. With this boolean mask for the boundary and the interior nodes, the incoming directions of the distribution functions are reversed when encountering a boundary node. This simple implementation further allows one to actually import images, convert them to black and white, and then model flow through the fluid regions. This setup was used to validate fluid flow through images found in the literature. The pitfalls of this approach is that complications arise at high Reynold's number and a high grid resolution is required to accurately model the system.

Two different boundary condition implementations were experimented on for the inlet/outlet boundary conditions; periodic boundaries and Zou-He boundaries. Although the equations for the periodic boundary condition are provided, in the interest of time only the Zou-He boundary condition was utilized for the rest of the experiments in this report.

Simple Bounce Back

The boundary condition was setup such that the fluid particles are reflected back into the fluid, when they hit the boundary. An on-grid bounce-back was implemented so that the distribution functions are reflected at the nodes at the boundary instead of propagating past the domain. Although the on-grid boundary was easier to implement, it has a first order of accuracy as compared to the mid-grid boundary, which has a second order accuracy.



For example at if a node is at a boundary with the fluid region to the right, the boundary condition is implemented such that in the i = 0,1,2,4,5,8 directions the probability functions are propagated normally but in the i = 3,6,7 directions the functions are reflected in the i = 1,5,8 directions respectively.

Periodic Boundary Condition

The periodic boundary condition was setup such that the propagated velocity and density functions at the outlet are transferred to the inlet to simulate Poiseuille channel flow.

$$\rho\,(\mathrm{x}+\mathrm{Li},\mathrm{t}) = \rho\,(\mathrm{x}+\mathrm{Li},\mathrm{t}) + \delta t \qquad \vec{u}\,(\vec{\mathrm{x}},\mathrm{t}) = \vec{u}\,(\vec{\mathrm{x}},\mathrm{t})$$

From this setup, a body force was applied to drive fluid flow. The body force was implemented such that at each time step, a fixed amount of momentum is added to all points within the fluid space. The body force was added to the collision operator such that:

$$\mathrm{f_i}\left(\vec{\mathrm{x}} + \vec{\mathrm{c}}_\mathrm{i},\ \mathrm{t}+1\right) = \frac{\mathrm{f_i}\left(\vec{\mathrm{x}},\ \mathrm{t}\right)(\tau-1) + \mathrm{f_i}^{eq}\left(\vec{\mathrm{x}},\ \mathrm{t}\right)}{\tau} + \mathrm{F}_{Body}$$

$$\mathrm{F}_{Body} = \frac{8 * \nu * U_f}{Ny^2}$$

$$\nu = \frac{UH}{Re}$$

Zou He Boundary Condition

Another method to simulate Poiseuille channel flow is to use Zou-He boundary conditions to impose Poisuelle Ux profile at Uy=0, the inlet, and Zou/He Constant Pressure Boundary at outlet all at each time step (Zou 1997). A pressure boundary condition was implemented at the outlet rather than another velocity boundary condition because a pressure boundary condition would avoid the effect of mass increase.

Thus, at the inlet assumed at x=0, the propagation step is as follows as *x* is varied from 0 to 1 and the velocity profile is applied:

$$\sum_{i=0,1,3,4,7,8} \mathrm{f_i}([\mathrm{x},1] + \vec{\mathrm{c}}_\mathrm{i}, \mathrm{t} + 1) = \frac{\mathrm{f_i}\,([\mathrm{x},1],\mathrm{t})(\tau - 1) + \mathrm{f_i}^{eq}\,([\mathrm{x},1]\,,\mathrm{t})}{\tau}$$

$$\rho = \frac{\mathrm{f}_0 + \mathrm{f}_2 + \mathrm{f}_4 + 2(\mathrm{f}_3 + \mathrm{f}_6 + \mathrm{f}_7)}{(1-u)}$$

$$\mathrm{f}_1' = \mathrm{f}_3 + \frac{2}{3}\rho u$$

$$\mathrm{f}_5' = \mathrm{f}_7 - \frac{1}{2}(\mathrm{f}_2 - \mathrm{f}_4) + \frac{1}{6}\rho u + \frac{1}{2}\rho v$$

$$\mathrm{f}_8' = \mathrm{f}_6 + \frac{1}{2}(\mathrm{f}_2 - 4) + \frac{1}{6}\rho u - \frac{1}{2}\rho v$$

Similarily, at the outlet assumed at x=L, a density of one was imposed:

$$\rho\,(\mathrm{L}\,,\mathrm{t}) = 1$$

4.7 CONVERGENCE CRITERIA

The steady state criteria implemented was that the norm of the difference between the current and previous *U* as well as *V* both had to be less than 10^{-4}:

$$\sqrt{\sum_{k=1}^{\#\ of\ elements}({U^{t+1}}_k - {U^t}_k)^2/N^2} < .0001 \Rightarrow \text{S.S.}$$

$$\sqrt{\sum_{k=1}^{\#\ of\ elements}({V^{t+1}}_k - {V^t}_k)^2}\ /N^2 < .0001 \Rightarrow \text{S.S.}$$

V. ORDER OF SCHEMA

Since all the equations involved a second order spatial discretization as explained in Section IV, the spatial convergence rate is expected to be 2.

The spatial convergence rate is found through the following equation for p, the order of the scheme:

$$p = \frac{\log(\frac{\varphi_{2h} - \varphi_{4h}}{\varphi_h - \varphi_{2h}})}{\log(r)}$$

Where φ_h is the stability criteria error metric on grid h and r is the grid refinement ratio; φ_{2h}/φ_h. Additionally, in the case of this analysis φ_h, φ_{2h}, and φ_{4h} correspond to solutions at a N of 100x100, 200x200, and 400x400 respectively.

For the Zou He boundary implementation of basic Poiseuille flow the calculated p was 2.5243, which is indeed very close to the expected convergence rate of 2. Presumably, if a half way bounce-back boundary condition was applied this rate could be improved.

VI. STABILITY

For the Lattice Boltzmann Method to be stable, the Mach number must be smaller than 1 and the ν, kinematic viscosity, must be greater than 0.

The Mach number is defined as follows:

$$Ma = \frac{u}{\sqrt{3}}$$

Since the Lattice Boltzmann method enters a slightly compressible regime, when simulating incompressible flows these compressibility effects must be minimized to preserve the accuracy of the solution. These compressibility effects scale with Ma^2, therefore the Mach number must be minimized. Thus, when one increases the grid resolution to improve numerical accuracy, they must also minimize the Mach number, which means reducing *u* (in lattice units ~ δt/δx), leading to the constraint that $\delta t \sim \delta^2 x$.

The kinematic viscosity and its relation to the relaxation time is defined as:

$$Re = \frac{UH}{\nu} = \frac{4N\widehat{U}}{\tau - \frac{1}{2}}$$

Thus to maintain positive viscosity, τ needs to be greater than .5.

At high Re, the Lattice Boltzmann Method with the BGK collision operator leads to asymptotic, oscillatory solutions and is numerically unstable. This is because the relaxation parameter becomes too small for high Re, low viscosity solutions as shown the τ equation reproduced from above.

$$\tau = \frac{1}{2}(6 * \frac{u_{lid}}{Re}\frac{\Delta t}{\Delta x^2} + 1)$$

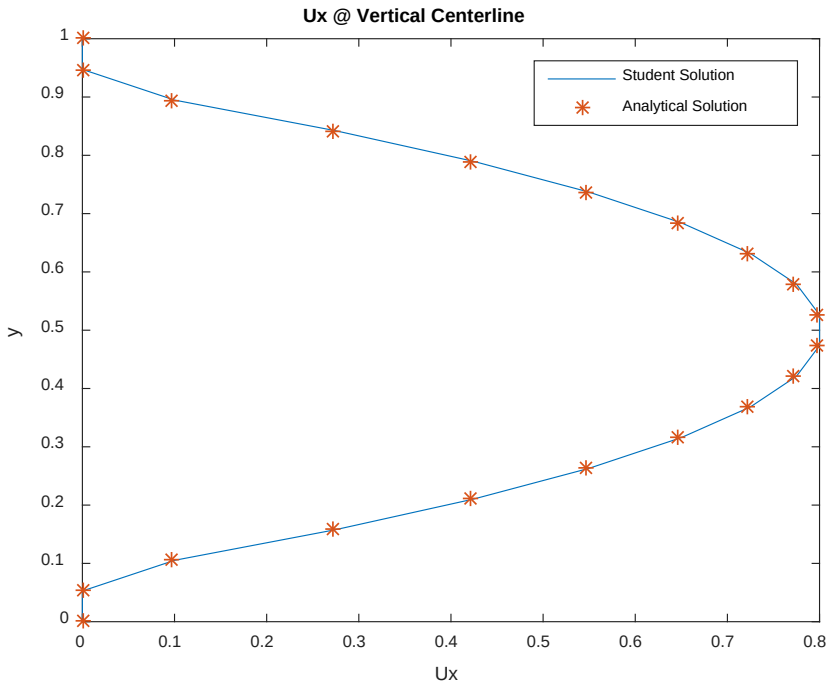
A smaller relaxation parameter leads to slower calculation of the collision operation.

VII. RESULTS AND VALIDATION

7.1 POISEUILLE FLOW

As explained above, this project was started with by implementing Poiseuille Flow through the use of Zou-He boundary conditions. In the base implementation, the horizontal velocity profile, U_y, is trivial because there are no flow obstructions. Rather to validate this model the U_x across the y-axis at the center of the flow was plotted along with the analytical solution calculated as:

$$U_x(y) = \frac{H^2}{4\mu}\left(-\frac{\partial P}{\partial x}\right)[1 - (\frac{y}{H})^2$$



7.2 FLOW AROUND CYLINDER

The student solution was benchmarked against the classic CFD problem of flow around cylinders. Flow around cylinders was accomplished by setting a circular area in the Immersed Boundary mesh to zero.

Additionally a Drag Coefficient to benchmark against the literature solution was accomplished as follows:

$$C_D = \mathrm{f_i}\left(\vec{\mathrm{x}}, \mathrm{t}\right)(\tau - 1) + \mathrm{f_i}^{eq}\left(\vec{\mathrm{x}}, \mathrm{t}\right)$$

Additionally, the Reynold's number equation also changes as follows for the flow past a cylinder problem:

$$Re = \frac{UD_{cylinder}}{\nu}$$

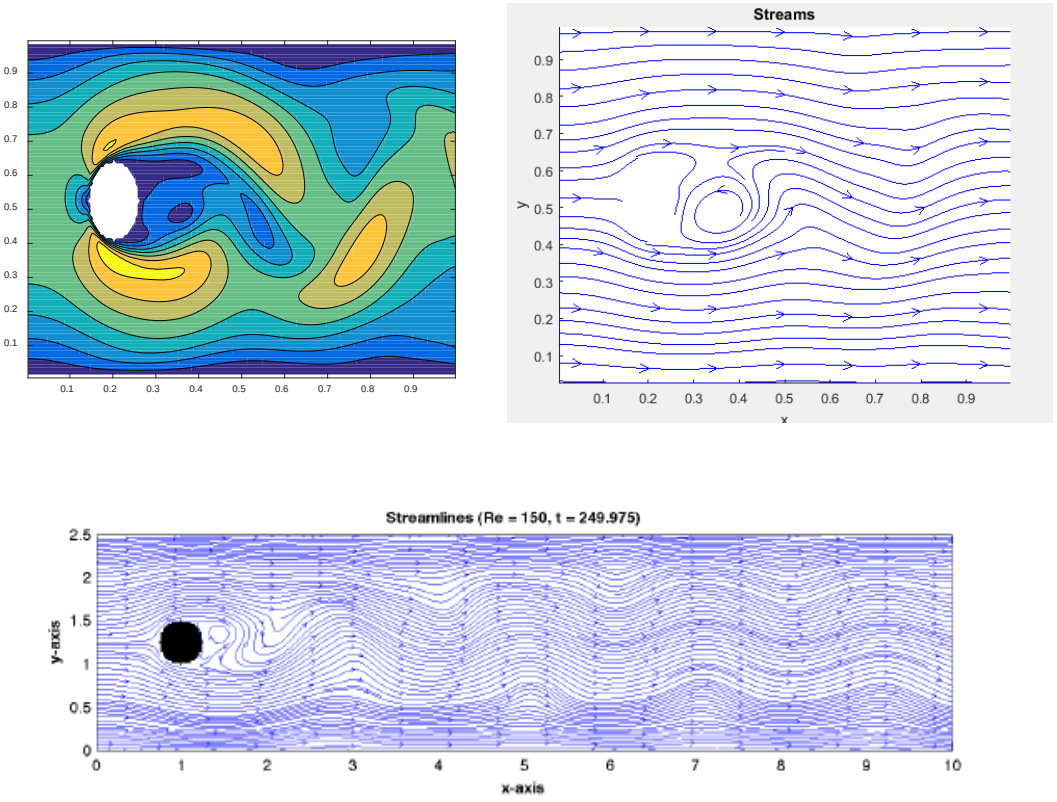


Figure 13: Streamlines of flow at $Re = 150$

The above plots were achieved at Reynolds of 150, which is in broad agreement with the literature solution provided underneath.

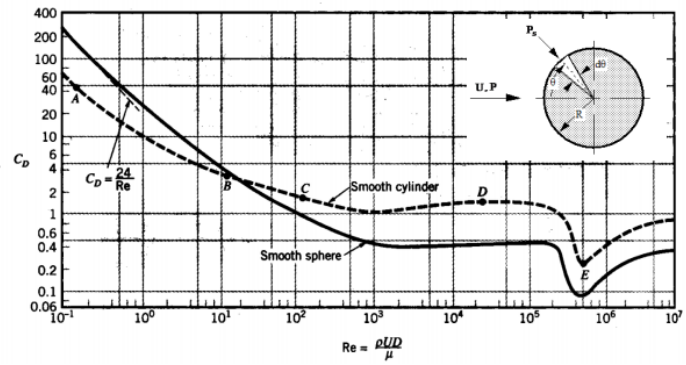
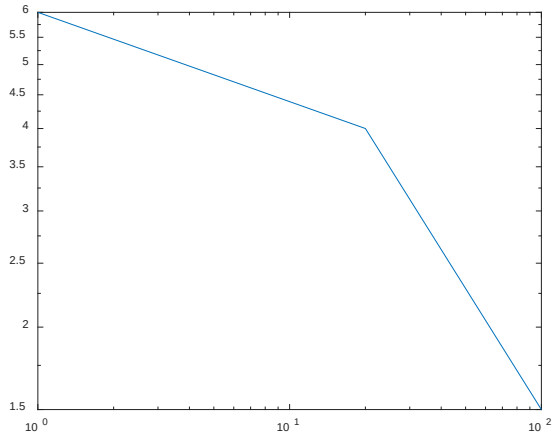


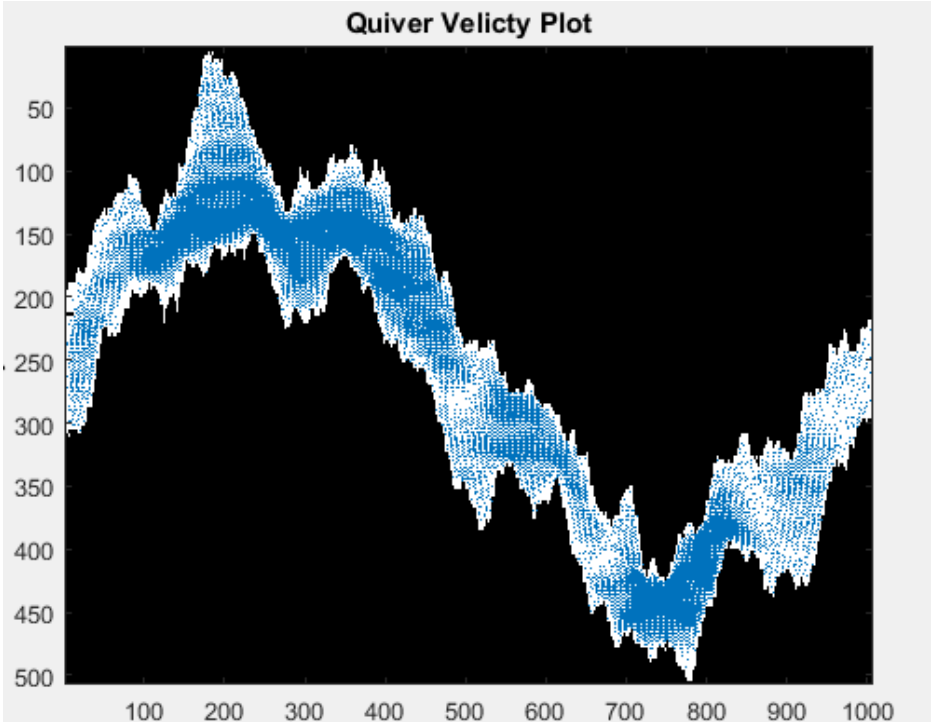
Fig. 2 Drag coefficient as a function of Reynolds number for a smooth circular cylinder (Adapted from Ref. [1])



Despite an extensive and thorough search through literature, no wake profile data points could be found. Thus, this Cd plot was utilized instead to validate the model. Higher Reynold's flow wasn't implemented due to a lack of time and because the drag coefficient was derived very late into the project.

7.3 FLOW THROUGH FRACTURE APERATURE

An application of Lattice Boltzmann fluid modelling is to use it to model fluid flow through cracks and apertures. This specific journal article analysed real rock fracture patterns and geometries through image recognition, used proprietary software to generate artificial fracture patterns and then modelled fluid flow through these fractures through Lattice Boltzmann Method. The Immersed boundary method implementation allows the user to read in images, convert them to black white and use this as a Boolean mask to determine fluid pathways.



Another example with the literature validation is provided below:

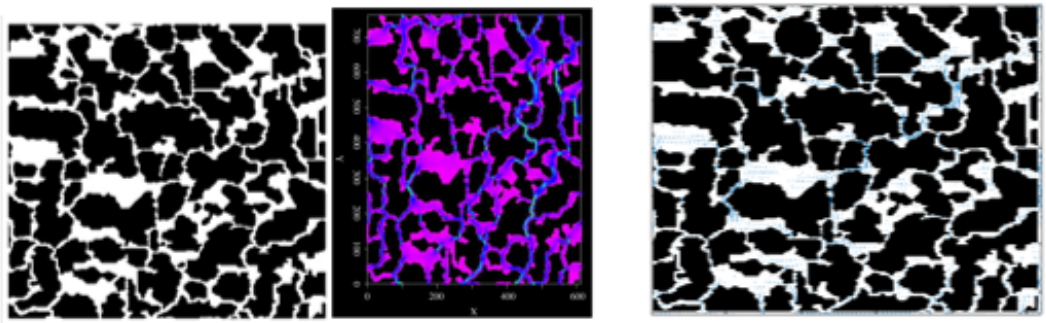
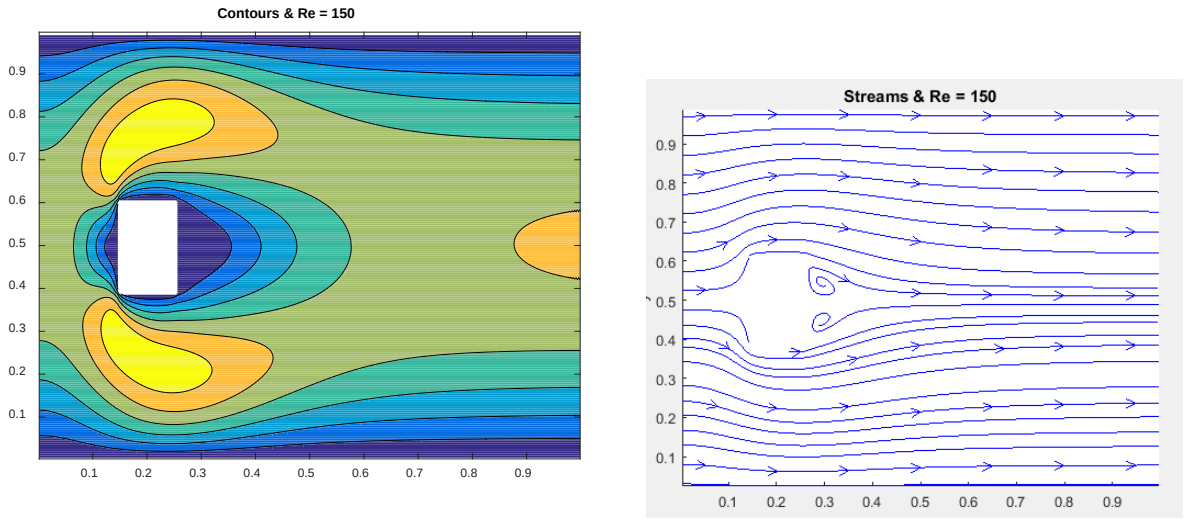
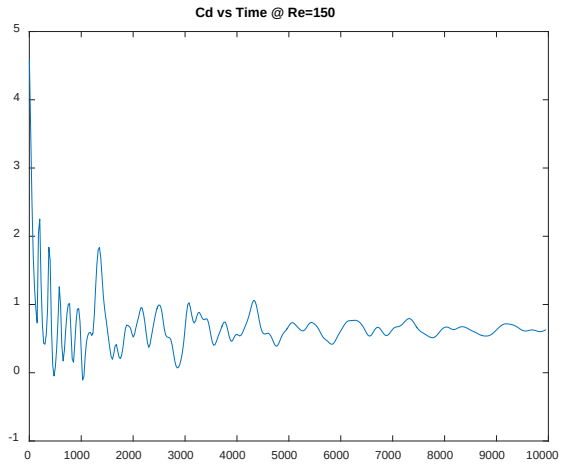


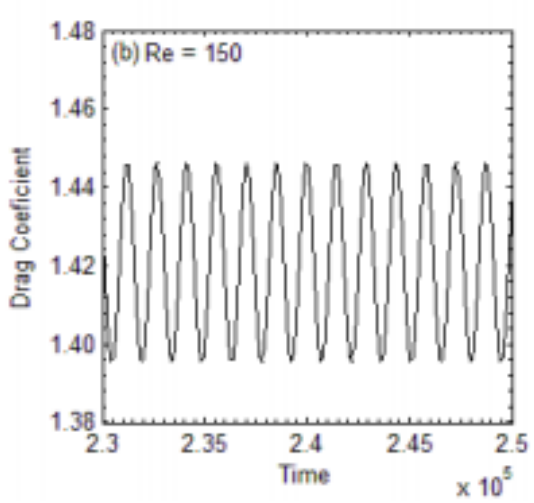
Fig. 1.
Berea pattern used in the lattice-Boltzmann 2D simulations. The obstacles are represented in black, and the pore space in white. The size is 1418 μm by 1774 μm. The etch depth of the physical Berea unit is 24.54 μm.

7.4 FLOW AROUND RECTANGULAR CYLINDERS



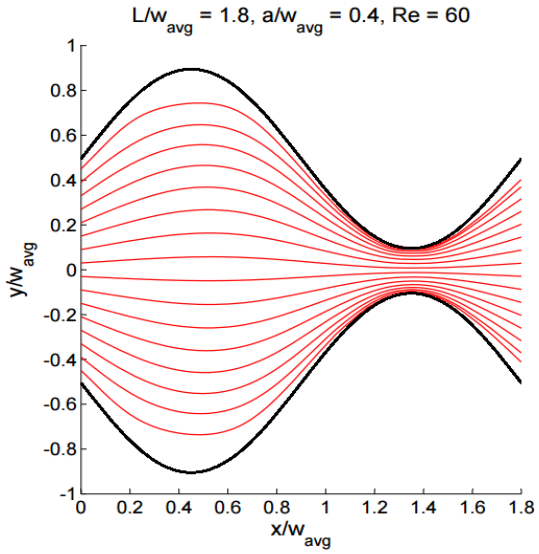
Since, the Immersed Boundary Method allows the user to modify the modelled object so easily. A quick example of flow around rectangular cylinders was accomplished.





The drag coefficient is broadly in the range of literature values although there are some discrepancies (Islam 2004).

7.5 FLOW THROUGH SINOSUIDAL PORE



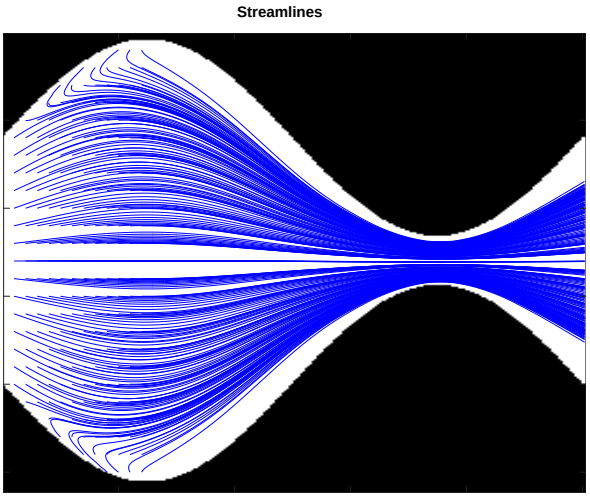


Figure . Sinosuidal Pore

The boundary mesh was implemented in such a way so as to create a sinosuidal pore throat as commonly found in rock geometries. When compared with the literature solution on the left, the student solution appears to be valid as well. The differences between the solutions is most likely due to the way the streamlines were plotted. The literature solution plots streamlines starting from the narrow pore throat. The student solution on the other hand didn't have that feature as the student streamline implementation could only follow the the direction the *u* velocity to the right.

VIII. ANALYSIS

Overall a variety of implementations were done to take advantage of the Immersed Boundary Method's high degree of flexibility and freedom. The solutions were implemented successfully without error and validated.

IX. CONCLUSION

The immersed boundary method combined with the channel flow allows a true exploration of all that CFD has to offer. A diverse range of fluid flows and obstacles were implemented and modelled, giving interesting results in the process.

X. REFERENCES

Shams-Ul-Islam, Waqas Sarwar Abbasi, and Hamid Rahman. "Force Statistics and Wake Structure Mechanism of Flow around a Square Cylinder at Low Reynolds Numbers."

Zou, Qisu, and Xiaoyi He. "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model." *Physics of fluids* 9.6 (1997): 1591-1598.

Munson, B.R., Young, D.F., and Okiishi, T.H., 2006, “Fundamentals of Fluid Mechanics,” 5th edition, John Wiley & Sons, p. 526.

XI. APPENDIX

See attached Matlab code.