Generalized Newtonian Fluid Simulations

Nitish Tripathi
Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500032
Email: nitish.tripathiug08@students.iiit.ac.in

P. J. Narayanan
Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500032
Email: pjn@iiit.ac.in

***Abstract*—We present an approach to simulate both Newtonian and generalized Newtonian fluids using Lattice Boltzmann Method. The focus has been on accurately modelling non-Newtonian fluids at the micro channel level from biological fluids in the past. Our method can model macroscopic behaviour of such fluids by simulating the variation of properties such as viscosity through the bulk of the fluid. The method works regardless of the magnitude of flow, be it through a thin tube or a large quantity of liquid splashing in a container. We simulate the change in viscosity of a generalized Newtonian fluid and its free surface interactions with obstacles and boundaries. We harness the inherent parallelism of Lattice Boltzmann Method to give a fast GPU implementation for the same.**

I. Introduction

Computational simulation of fluids is very important for several applications. The incompressible Navier Stokes' equations [1] describe the physics behind fluids. They are a set of two partial differential equations which hold throughout the fluid and model it on the basis of Newton's second law of motion.

$$\frac{\partial u}{\partial t} + \vec{u}.\nabla\vec{u} + \frac{1}{\rho}\nabla p \;=\; \vec{g} + \nu\nabla.\nabla\vec{u} \qquad (1)$$

$$\nabla.\vec{u} = 0 \qquad (2)$$

Eq. (1) is called the *momentum equation*. This, essentially, is Newton's second law of motion. u is the velocity field, ρ is the density, p is the pressure existing in the fluid, g is the acceleration due to gravity while ν is the kinematic viscosity. Eq. (2) is called the *incompressibility condition* and is interpreted as the amount of fluid entering and exiting is the same through any element of unit volume. The two equations together describe the velocity field through the fluid domain completely.

The non-linearity of the first equation makes it difficult to solve analytically, with non-existent solutions for flows with higher Reynold's number. However, exact solutions to the problem exist due the vanishing of the non-linear term in such cases.

We generally take the Newtonian fluid model as standard. However, Newtonian fluid is just one of the many kinds of fluid behaviour which we see around us. An example of a non-Newtonian fluid is blood. It is a shear-thinning fluid, for which the apparent viscosity decreases with the increase in shear rate. Other kinds such as shear-thickening fluids also exist. Other examples of such fluids are multiphase mixtures like slurries and emulsions, polymer and solutions, food products like jams and jellies etc.

Conventionally, two viewpoints have existed for modelling fluids viz. Lagrangian and Eulerian. While both methods involve directly discretizing the Navier Stokes' equations, the Lagrangian viewpoint [2] is the more intuitive of the two. It models the fluid as a particle system. Each point is marked with position and velocity vectors. It tracks these points and models how the vectors change according to the variation in fluid properties such as density, velocity and temperature. Eulerian approach [3] takes the opposite direction; it looks at fixed points in the fluid domain and calculates how fluid properties change on these points as it flows past them.

Another approach, which is relatively new, is the Lattice Boltzmann Method (LBM). It is based on Kinetic Theory. The LBM evolved from Lattice Gas Cellular Automata when the latter was found to be incapable in simulating fluids with high accuracy. The method takes a mesoscopic view of the problem showing that modelling at this level can aggregate to give an accurate description of macroscopic fluid properties. The model, thus, does away with the need to solve partial differential equations or apply approximate finite difference techniques and is extremely easy to understand and implement. Also, it is accurate to the second order, in contrast to other methods which are accurate only to the first order. Although several sub-approaches exist we will focus on LBM with Bhatnagar, Gross, Krook (BGK) approximation [4].

In the past attention has been paid to simulating non-Newtonian behaviour at extremely small scales. Flow of blood in capillaries is an example of the kind of flows that has been studied. However, we argue that since the physics behind the behaviour of fluids doesn't change according to the scale or extent of flow, the same equations can be used to model *bulkier flows*. We do this by reducing dependence of the model on constants like the *Knudsen number*. The Knudsen number is the ratio of the molecular mean free path to the characteristic physical length scale of the simulation. While it is easy to find the characteristic length for microfluidics and is taken to be the width of the channel (for example, the characteristic length of a tube is it's diameter), it becomes difficult to find its value for simulations where there is no channel to speak of, for example, a fluid flowing on an inclined plane. In the succeeding sections we introduce the theory behind LBM and how it can be extended. The procedure can model generalized Newtonian flows of any scale, implying that Newtonian, shear-thinning and dilatant fluids can be simulated and the change in viscosity depicted, not just in the bulk of the fluid but also as its *free surface* interacts with obstacles, boundaries and atmosphere while progressing forward.

II. Generalized Newtonian Fluids

The Newtonian or non-Newtonian nature of a fluid can be determined by its flow curve i.e the plot of shear stress vs. shear rate. The slope of the curve gives the viscosity of the fluid. If the slope is constant and the curve passes through the origin the fluid is Newtonian. In other words, for Newtonian fluids, viscosity is independent of shear rate and depends only on temperature and pressure. In addition to that, for an incompressible Newtonian flow, the deviatoric normal stresses should be zero in simple shear i.e.

$$\tau_{xx} \;=\; \tau_{yy} \;=\; \tau_{zz} \tag{3}$$

On the other hand if the curve is non-linear or, linear but not passing through the origin then the fluid is said to be non-Newtonian. The non-Newtonian behaviour falls under three categories viz.

1) Fluids in which shear stress depends only on the current value of the shear rate. $\tau_{yx} = f(\dot{\gamma_{yx}})$.
2) Fluids where the shape of the flow curve depends on the duration of shearing etc.
3) Materials which show partial elasticity after deformation thereby exhibiting combined characteristics of both an elastic solid and a viscous fluid.

The first kind of fluids are called Generalized Newtonian Fluids (GNF) and it is this type that we will be focussing on in this paper. Thus Newtonian fluids are a specific case of GNF.

III. The Lattice Boltzmann Method

The LBM [4] in its most basic form consists of two steps, advection (or streaming) and collision. It is a cellular automata based approach. The simulation domain is divided into grid cells with each cell directly interacting only with its neighbours. The lattice is typically named DXQY where X is the number of dimensions of the domain and Y determines the number of directions the particles may move to from a cell site. Thus the method restricts the number of distinct lattice velocity directions since each lattice velocity vector points to a neighbouring cell. D2Q9 and D3Q19 grids are most popular for 2 dimensional and 3 dimensional simulations respectively.

Like LGCA, each cell stores the number of particles that move along each lattice velocity vectors. Only, these are real numbers unlike in LGCA where they are boolean, and are called *particle distribution functions*. They are represented as f_i with i being the number of lattice velocity vector. Thus summing up all the particle distribution functions for a cell gives the total number of particles in the cell at that time instant. Since the cell size is assumed, for ease of simulation, to be unity, this is also its density. Momentum density can be calculated by taking a weighted average of the velocity vectors using the *pdfs* as weights. This procedure is called *course graining*. Hence,

$$\rho \;=\; \sum_i f_i \tag{4}$$

$$\rho u \;=\; \sum_i f_i * e_i \tag{5}$$

Where e_i are the lattice velocity vectors.

In the advection step the $f_i s$ are copied to the neighbouring cell according to the lattice velocity vectors. As f_0 doesn't point anywhere (it is equal to the number of particles that are stationary within a cell) its value for each grid cell remains the same post the stream step.

The collision step imitates the collisions of the fluid particles occurring in the simulation domain. For colliding, the equilibrium distributions are calculated first based on the lattice velocity vectors and velocity of the cell,

$$f_i^{(0)} \;=\; w_i * [\rho - 3/2(\overrightarrow{u})^2 + 3(\overrightarrow{e_i}.\overrightarrow{u} + 9/2(\overrightarrow{e_i}.\overrightarrow{u})^2] \tag{6}$$

where w_i are the weights awarded to each direction depending on the type of grid (*D3Q19*, *D2Q9* etc). The relaxation time τ is linked to the kinematic viscosity ν through,

$$\nu \;=\; (2 * \tau - 1)/6 \tag{7}$$

The new particle distribution functions after the collide step according to BGK approximation [4] are given by,

$$f_i' \;=\; f_i * (1 - 1/\tau) + f_i^{(0)} * 1/\tau \tag{8}$$

It is assumed that f_i obtained for the next step can be expanded around the local equilibrium distribution function [5] i.e.

$$f_i \;=\; f_i^{(0)} \;+\; \epsilon * f_i^{neq} \tag{9}$$

Where ϵ is the *Knudsen number* [6] and f_i^{neq} is the non equilibrium part of the distribution function expanded as,

$$f_i^{neq} \;=\; f_i^{(1)} \;+\; f_i^{(2)} \;+\; o(\epsilon^2) \tag{10}$$

$$\text{where,} \quad \sum_i f_i^{(k)} \;=\; \sum_i f_i^{(k)} * e_i \;=\; 0, \qquad k \;=\; 1, 2. \tag{11}$$

After obtaining the distribution functions at the end of the stream and collide steps one can coarse grain to find the value of density and velocity at each grid cell location. The obtained values are in accordance with the Navier Stokes' equations and are accurate to second order.

As described above the relaxation time is related to the kinematic viscosity. For a Newtonian fluid this isn't an issue since the viscosity is constant for given temperature and pressure. However for a Generalized Newtonian Fluid this may not be true. We look at GNF simulations based on the truncated *Ostwald-de Waele* model, also known as the *truncated* power law model [7] ,

$$\nu \;=\; \begin{cases} k * \dot{\gamma}_0^{n-1} & \dot{\gamma} < \dot{\gamma}_0 \\ k * \dot{\gamma}^{n-1} & \dot{\gamma}_0 < \dot{\gamma} < \dot{\gamma}_\infty \\ k * \dot{\gamma}_\infty^{n-1} & \dot{\gamma} > \dot{\gamma}_\infty \end{cases} \tag{12}$$

where $\dot{\gamma}$ is the shear rate. We use the truncated model because many fluids display non-Newtonian behaviour only in some range shear rates; outside that range they behave as Newtonian fluids. When $\dot{\gamma}$ lies in that range, fluid is shear-thinning if $0 \leq n < 1$, shear-thickening if $n > 1$ and Newtonian with viscosity $\nu = k$ otherwise. Another reason for using the truncated power law is the LBM becomes unstable as $\nu \to 0$ or $\nu \geq 1/6$.

The viscous stress tensor for a fluid d is defined as,

$$d \;=\; 1/2 * (\nabla u + \nabla u^t) \tag{13}$$

It has been shown by [8] that the viscous stress tensor can be recovered at each node by using the relation,

$$d(x,t)_{\alpha,\beta} = -3/(2*\tau)*\sum_i f_i^{(1)}(x,t)e_{i\alpha}e_{i\beta} \tag{14}$$

However eq. (14) requires $f_i^{(1)}$ which is obtained using eq. (9). However we choose to employ the more fundamental relation i.e. eq. (13). We do this for two reasons, first, because its only a question of applying finite difference technique to calculate ∇u and ∇u^t after the calculation of instantaneous velocities. The velocity field is anyway calculated using coarse graining for each frame. Second, knowledge of the *Knudsen number* which may vary along with the scale of simulation. For example, it will be different for flows in a thin tube and for *bulkier flows* like curry in a pot. Also, we realize that applying finite difference will restrict the accuracy of our method to the first order whereas Lattice Boltzmann equations are second order accurate. However, this is not an issue since using boundary conditions like *no-slip* make the overall simulation first order accurate.

The second invariant of the strain rate tensor [5] is,

$$D_{II} = \sum_{\alpha,\beta=1}^{dimension} d_{\alpha,\beta}d_{\alpha,\beta}, \tag{15}$$

Where $dimension$ is the dimensionality of the grid. For two dimensions, this is equal to $det(d)$. The shear rate can then be obtained from the above as,

$$\dot{\gamma} = 2\sqrt{D_{II}} \tag{16}$$

We obtain the new value for viscosity using eq. (12). Putting this value in (7) we find the shear dependent τ for the next time step.

Modelling Free Surfaces

We extend the process by simulating the interaction of fluid with their environment. We use *no-slip boundary conditions* to model the interaction of the fluid with the container boundaries. To model its interaction with the atmosphere we use the procedure given by [9]. The cells which are full of fluids are treated the same way as above, while special treatment is given to the cells lying on liquid-air interface which are partially filled with fluid, as given in their algorithm. The distribution functions for such cells need to be normalised based on the type of cells in their neighbourhood (liquid, air or interface).

IV. Parallel Implementation

The parallelism in LBM is evident from the above description. As it proceeds by aggregating quantities in a bottom up fashion and starts from a mesoscopic scale there are extremely few computationally heavy calculations. We harness these features while accelerating it using the GPU. Both the stream step and the collide step consist of local operations which make the computations of each cell independent of one another. This facilitates direct mapping of these operations to parallel kernels. Data access patterns form an important factor in the performance of the simulations and hence we try to access data in an optimized manner. We employ a *SoA*(Structure of Arrays) data format to store information present in each cell which enables coalesced memory access for threads. In our simulations the entire simulation domain is considered as a grid made up of blocks. $32x32$ cells make up a block, with a single thread handling a cell.

The stream step for a particular cell requires data from all its neighbouring cells. Data accesses to neighbouring cells on a per thread basis will result in lot of memory traffic and may lead to non-coalesced access patterns. This approach is thus avoided. Instead we make each thread load its contents into the shared memory allocated for the block. Since shared memory access is much faster than global memory accesses threads now can access their neighbouring cells' data faster thereby leading to performance gains. Threads handling cells at the block boundary have to be taken care of specially since the data required by them is not completely available in shared memory. This is handled as a special case within the kernels. Since collision requires each thread to work with only the data present with the cell, each thread is independent of the other and such issues don't arise.

V. Results and discussion

We show here the results of two standard experiments that are performed for bulk fluid simulations, the lid driven cavity experiment and the dam-break experiment. The intermediate frames from the simulations are shown as results in figures 1 and 2 respectively. We also show flow of a shear-thinning fluid in a thin tube with bifurcations to show that the method works well both of bulk as well as micro-simulations.

In the lid driven cavity experiment, a sealed container filled with fluid to the brim is simulated. The idea is for one wall to move along the domain accelerating the fluid with no slip boundary conditions. A special kind of cells called accelerator cells are marked which are awarded a fixed velocity. We keep the last but one row of our domain as accelerator cells implying that the top wall of the container is moving. After some time, a vortex becomes visible near the middle of the domain as can be seen in the frames. The color gradient present in the frames shows the variation of viscosity in the bulk of the fluid. The viscosity of the region decreases as the color shifts from red to blue to green. As is expected, the last but one row of the domain, which is driving the vortex has the most viscosity followed by areas around it which are coloured, successively, yellow, blue, magenta and red. The farther a cell from the accelerator row, the lesser is its viscosity.

Figure 2 shows intermediate frames from a dam break experiment with shear-thickening fluid. A wedge shaped obstacle is kept at three fourths of the length of a side of the domain to interact with the fluid. The experiment is important because it not only shows how the viscosity changes within the fluid, but also the change in viscosity as it progresses and its free surface interacts with the boundaries and the atmosphere. The gradient again shows the same variation in viscosity with red signifying maximum viscosity followed by blue and green. The motion of the fluid away from a wall implies a shear acting on the fluid cells close to the wall because of the no-slip boundary conditions resulting in increasing viscosity in the region as can be seen when the fluid is receding from the left wall. Also, the viscosity increases on the bottom layer of the fluid when it advances and occupies new volume. In figures 2a and 2b as the

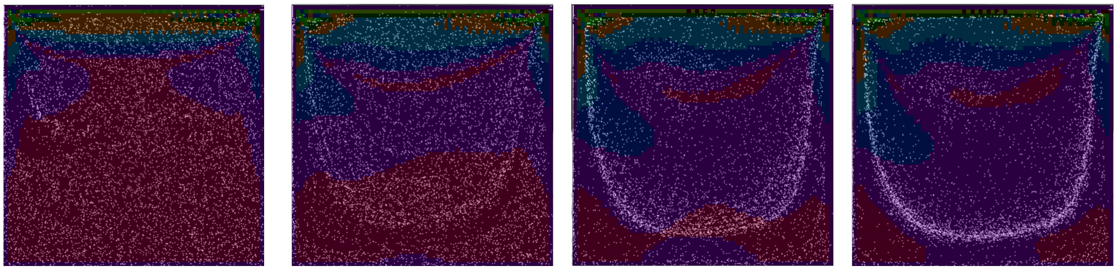
(a) (b) (c) (d)

Fig. 1: Intermediate frames of the ”Lid driven cavity” experiment

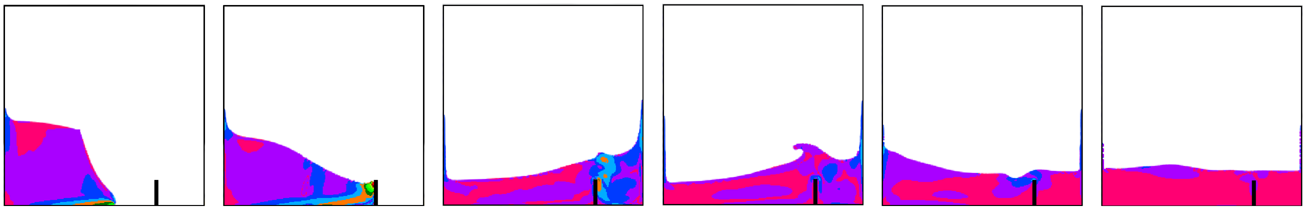
(a) (b) (c) (d) (e) (f)

Fig. 2: Intermediate frames of ”Dam-break” experiment simulation

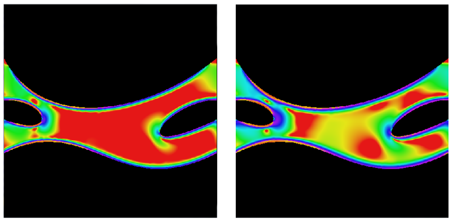
(a) (b)

Fig. 3: Shear-thinning flow in a thin tube

fluid advances and interacts with the vertical wedge there is a change in the color gradient from top (red) to bottom (from blue to orange to green) signifying the increase in viscosity in this direction.

Figure 3 shows viscosity change along a thin tube with shear-thinning non-Newtonian fluid. The simulation is done with no-slip boundary conditions to ensure zero fluid velocity along the tube walls. Here blue signifies the minimum viscosity, which can be found near the tube walls and the bifurcations as these are the regions of maximum stress. The tube center is the area experiencing minimum shear and correspondingly, maximum viscosity. The green areas are intermediate. Figure 3a experiences an overall lower shear stress rate than figure 3b, which is depicted by the shrinking of the (red) regions having comparatively high viscosity.

We have performed experiments with both shear-thinning and shear-thickening fluids here and have accrued accurate results for all. Further work includes making the method accurate enough to be used as efficient design and optimization tools in medical applications. The non-Newtonian nature of such fluids is more pronounced at finer levels. Microfluidic devices thus need extremely accurate representation because of their usage in biomedical research. Many of the complex flow phenomenon may not be easily obtained by physical experimentation. We would further like to extend this approach to simulate interaction of non-Newtonian fluids with moving obstacles and boundaries. We would therefore like to incorporate other factors as well, such as, channel geometry, diffusion coefficients, possible chemical interactions and particulate suspensions etc.

References

[1] V. Girault and P.-A. Raviart, “Finite element approximation of the navier-stokes equations,” *Lecture Notes in Mathematics, Berlin Springer Verlag*, vol. 749, 1979.

[2] M. Muller, D. Charypar, and M. Gross, “Particle-based fluid simulation for interactive applications,” in *Symposium on Computer Animation*, ser. SCA ’03, 2003, pp. 154–159.

[3] N. Foster and D. Metaxas, “Realistic animation of liquids,” *Graphical Models and Image Processing*, vol. 58, no. 5, pp. 471 – 483, 1996.

[4] S. Chen and G. D. Doolen, “Lattice boltzmann method for fluid flows,” *Annual Review of Fluid Mechanics*, vol. 30, no. 1, pp. 329–364, 1998.

[5] J. Boyd, J. Buick, and S. Green, “A second-order accurate lattice boltzmann non-newtonian flow model,” *Journal of Physics A: Mathematical and General*, vol. 39, no. 46, p. 14241, 2006.

[6] T. N. Phillips and G. W. Roberts, “Lattice boltzmann models for non-newtonian flows,” *IMA Journal of Applied Mathematics*, vol. 76, no. 5, pp. 790–816, 2011.

[7] S. Gabbanelli, G. Drazer, and J. Koplik, “Lattice boltzmann method for non-newtonian (power-law) fluids,” *Phys. Rev. E*, vol. 72, p. 046312, 2005.

[8] A. Artoli, A. Monim, M. Ali, and M. H. Artoli, “Mesoscopic computational haemodynamics,” 2003.

[9] N. Threy and U. Rde, “Stable free surface flows with the lattice boltzmann method on adaptively coarsened grids,” *Computing and Visualization in Science*, vol. 12, no. 5, pp. 247–263, 2009.