Numerical Methods Term Paper Report

Lattice Boltzmann Method to Understand fluid flows

Yuvraj Sharma

Astrophysical Relativity group, ICTS-TIFR Bangalore

Course Instructor

Prof. Parameswaran Ajith

Astrophysical Relativity group, ICTS-TIFR Bangalore

Course Instructor

Prof. Prayush Kumar

Astrophysical Relativity group, ICTS-TIFR Bangalore

Abstract

We simulate fluid flow past various obstacles using the Lattice Boltzmann method. This is a simple technique to simulate fluid flow, where instead of evolving the fluid (Navier-Stokes) equations directly, microscopic particles on a lattice are simulated with streaming and collision processes. The power of the method comes from reducing the high-dimensionality of the microscopic physics onto the lattice, which has limited degrees of freedom. We implement this method alongside various constraints (obstacles here) in a 2D lattice, and interpret the results in terms of magnitude of Velocity at every lattice point.

1 Introduction

The Lattice Boltzmann method (LBM) is a powerful computational technique used to simulate fluid flows. It is based on microscopic models of fluid dynamics and offers several advantages over traditional computational fluid dynamics (CFD) methods, such as its simplicity, scalability, and ability to handle complex geometries. LBM simulates fluid flow by modeling the motion of individual particles on a lattice grid. These particles follow simplified collision and propagation rules, which are derived from the Boltzmann equation in kinetic theory.

We start with discretizing the fluid domain into a lattice grid, where each lattice site represents a fluid particle. Various lattice models, such as D2Q9 [for 2D] and D3Q27 [for 3D], are used depending on the dimensionality and complexity of the flow. In the collision step, particles interact with each other according to collision rules that conserve mass, momentum, and energy. After collisions, particles propagate to neighboring lattice sites based on predefined velocity vectors. This step ensures that fluid particles move throughout the domain. By averaging the distribution of particles over each lattice site, macroscopic fluid properties such as velocity, density, and pressure are obtained. These variables provide insight into the behavior of the fluid flow. Boundary conditions are applied to the lattice grid to model physical boundaries and interactions with solid surfaces. Various boundary conditions, such as bounce-back and extrapolation, are used depending on the application.

2 LBM Solver Algorithm

In the following section, we will show the algorithm that solves the system with obstacles and gives us a representation of the 2D system with flowing fluid.

2.1 Microscopic Dynamics

LBM is rooted in kinetic theory, which describes to us the behavior of particles at the microscopic level. Here, fluid flow is simulated by modeling the movement of individual particles on a discrete lattice grid and each lattice site represents a fluid particle, and the state of the particle (e.g., velocity and density) is described by a distribution function called the Lattice Boltzmann distribution function (or a Phase space). We will use $f(\mathbf{x}, \mathbf{v})$ to describe the density phase, $\mathbf{x}$ represents the position vectors and $\mathbf{v}$ represents the velocity vector for every lattice points within the system.

2.2 Lattice

Within the fluid domain, each grid point corresponds to a node in the computational domain, which is a matrix containing lattice point locations and velocities, we represent using F. The choice of lattice grid depends on the dimensionality of the flow (2D or 3D) and the desired level of accuracy. Some common lattice models include D2Q9 (two dimensions, nine velocities) and D3Q27 (three dimensions, 27 velocities), and we will be using D2Q9 for our project.

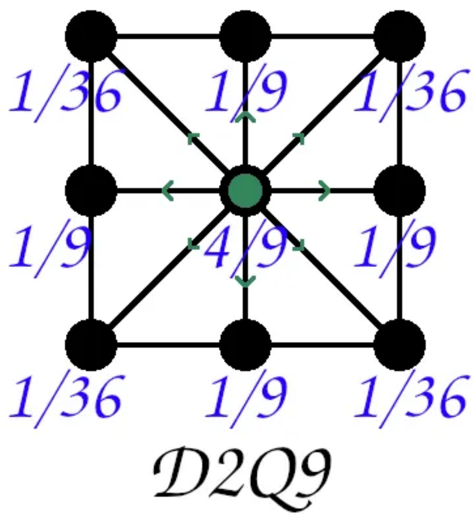

Figure 1: Velocity probability distribution of D2Q9 [5]

D2Q9 is a 2 dimensional lattice with 9 possible velocities at each lattice site (D2Q9). There are 4 connections running North, South, East, and West, 4 diagonal connections, and 1 connection from a node to itself representing zero velocity. Each lattice site also has a weight w_i associated with it, as shown in Fig. 1. Using the phase space density of fluid particles $f(\mathbf{x}, \mathbf{v})$ at location $\mathbf{x}$ with velocity $\mathbf{v}$. The particles will have the above shown probabilities to distribute the velocities to their adjacent lattice points.

2.3 Collision and Streaming

Within the flow, the particles will be mainly performing the following two actions, **streaming** and **collisions**. In the collision step, particles interact with each other according to simplified collision rules, and these rules help with the constraints to conserve mass, momentum, and energy, ensuring that the macroscopic fluid properties are accurately represented. The collision step is responsible for simulating viscosity and other fluid properties. Following collisions, the particles propagate to neighboring lattice sites based on predefined velocity vectors, this ensures that fluid particles move throughout the domain according to their velocities.

The streaming step is crucial for maintaining the continuity of fluid flow and accurately capturing fluid dynamics phenomena. This coupled behaviour can be represented using the BGK approximation:

$$(\partial_t + \mathbf{v}.\nabla)f = -\frac{f - f^{eq}}{\tau} \tag{1}$$

where, the left-hand side represents streaming, and the right-hand side approximates collisions. In this approximation, τ is the timescale of which collisions happen, and the distribution function f tends towards some equilibrium state f^{eq} as a result. We can discretize the equation onto our lattice as follows,

$$F_i(\mathbf{x}_i + \mathbf{v}_i\Delta t, t + \Delta t) - F_i(\mathbf{x}_i, t) = -\frac{F_i(\mathbf{x}_i, t) - F_i^{eq}(\mathbf{x}_i, t)}{\tau} \tag{2}$$

where, i denotes each of the 9 lattice directions with velocity $\mathbf{v}_i$

The first step in the LBM is to stream the particles. Conceptually, at each lattice site, for each direction i, the value F_i is shifted over to the neighboring lattice site along the connection. Typically this method uses units of $\Delta t = \Delta x = 1$ and we will use this convention throughout. The streaming velocities are hence: (0,0), (0,1), (0,-1), (1,0), (-1,0), (1,1), (1,-1), (-1,1), (-1,-1) for each of the 9 directions as shown in Fig. 1. Next we need to define the equilibrium state as a result of collisions. This depends on the fluid model's equation of state. For our system, we have assumed an isothermal (constant temperature) fluid, which has a constant sound speed. We define units using common conventions such that the lattice speed is $c = 1$ (which corresponds soundspeed = 1/3). The equilibrium state is given by:

$$F_i^{eq} = w_i\rho\left(1 + 3(\mathbf{v}_i \cdot \mathbf{u}) + \frac{9}{2}(\mathbf{v}_i \cdot \mathbf{u})^2 + \frac{3}{2}(\mathbf{u} \cdot \mathbf{u}\right) \tag{3}$$

which corresponds to the isothermal Navier-Stokes equations with a dynamic viscousity:

$$\mu = \rho\left(\tau - \frac{1}{2}\right)\Delta t \tag{4}$$

2.4 Macroscopic Variables

We average the distribution of particles over each lattice site and the macroscopic fluid properties such as velocity, density, and pressure are obtained. These macroscopic variables provide valuable insights into the behavior of the fluid flow and can be used to analyze and visualize flow patterns and phenomena within our 2D lattice. For this, the moments of the discrete distribution function can be taken to recover fluid variables at each lattice site. With this, we obtain the density:

$$\rho = \sum F_i \tag{5}$$

and momentum:

$$\rho\mathbf{v} = \sum F_i\mathbf{v}_i \tag{6}$$

where the sum is over all lattice directions, which here for our 2D lattice is 9.

2.5 Boundaries

Boundary conditions are applied to the lattice grid to model physical boundaries and interactions with solid surfaces. Different types of boundary conditions, such as bounce-back and extrapolation, are used depending on the specific application and the nature of the boundary. Boundary conditions play a crucial role in accurately representing the behavior of fluid flow near solid surfaces and boundaries. Here, boundary conditions are implemented on the microscopic level. In our simulation, we wish to add a solid obstacle, and lattice sites part of this obstacle will be flagged. At these flagged locations, particles will behave differently, we will consider reflective boundary conditions. Instead of collisions that lead to equilibrium, particles will simply bounce back. This is easily accomplished by swapping lattice directions:

$$F_i \longleftrightarrow F_j \tag{7}$$

where i and j correspond to lattice directions that point in opposite directions.

3 Results

Our code uses the above algorithm and starts by setting up the lattice and initial condition for F_i, and alternates streaming and collision (+boundary) operators to evolve the system. We notice in the resulting images that this restricted microscopic representation is able to capture macroscopic fluid behavior. The initial conditions place our obstacle into a periodic box with rightward moving fluid. As the flow progresses, turbulence develops in the wake behind the obstacles, and these are known as the Kármán vortex streets. In our visualizations, we represent the magnitude of velocities at every lattice point, with lighter shade representing higher velocity magnitudes and darker shade representing lower velocity magnitudes. Following are the various obstacles we used for our simulations and obtained the visualizatons:

3.1 Planar Sheet

3.1.1 Reynolds number: 20

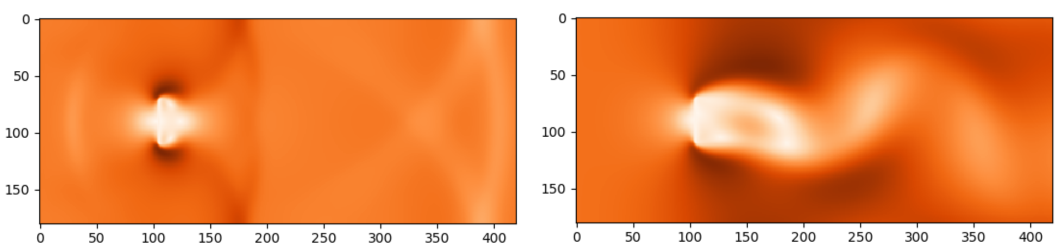

Figure 2: Obstacle: Sheet — Reynolds number: 20

3.1.2 Reynolds number: 1100

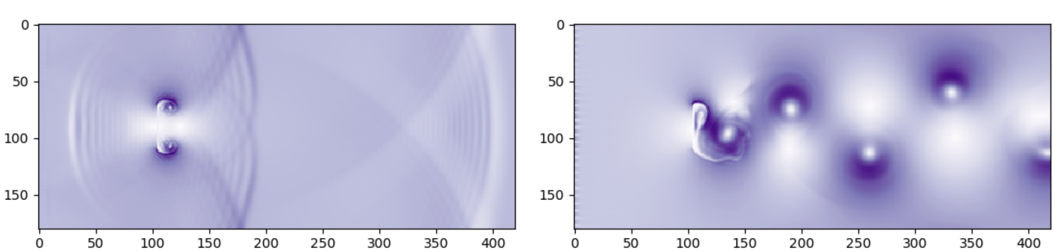

Figure 3: Obstacle: Sheet — Reynolds number: 1100

3.2 Circle

3.2.1 Reynolds number: 20

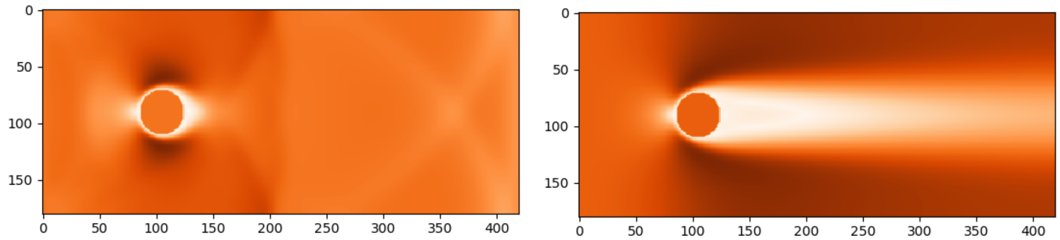

Figure 4: Obstacle: Circle — Reynolds number: 20

3.2.2 Reynolds number: 90

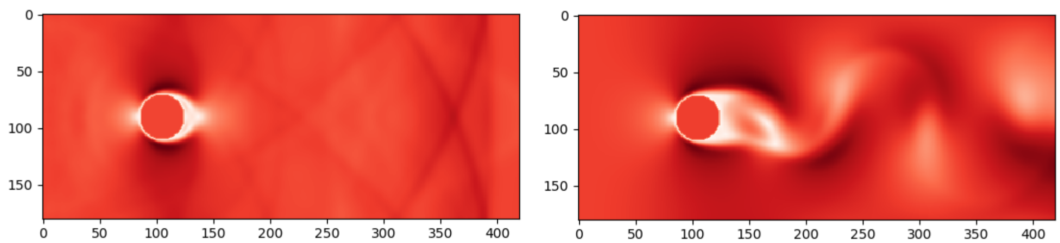

Figure 5: Obstacle: Circle — Reynolds number: 90

3.2.3 Reynolds number: 1100

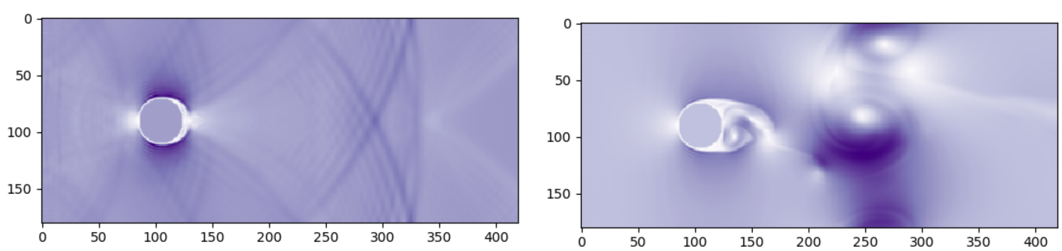

Figure 6: Obstacle: Circle — Reynolds number: 1100

3.3 Triangle

3.3.1 Reynolds number: 20

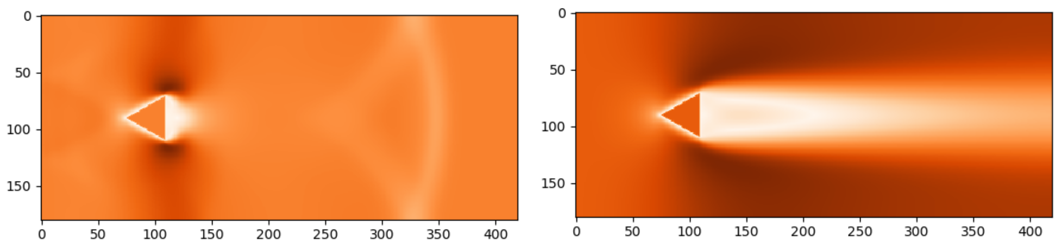

Figure 7: Obstacle: Triangle — Reynolds number: 20

3.3.2 Reynolds number: 90

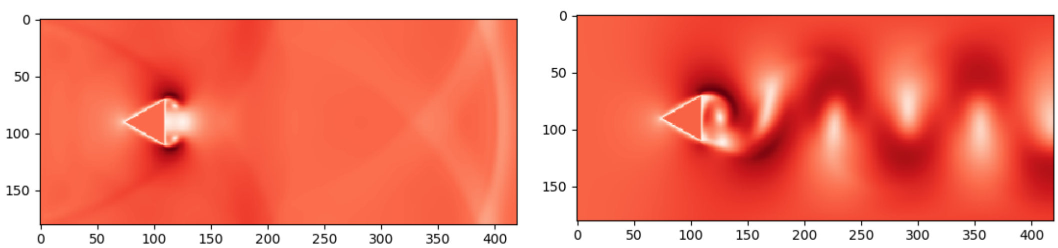

Figure 8: Obstacle: Triangle — Reynolds number: 90

3.3.3 Reynolds number: 1100

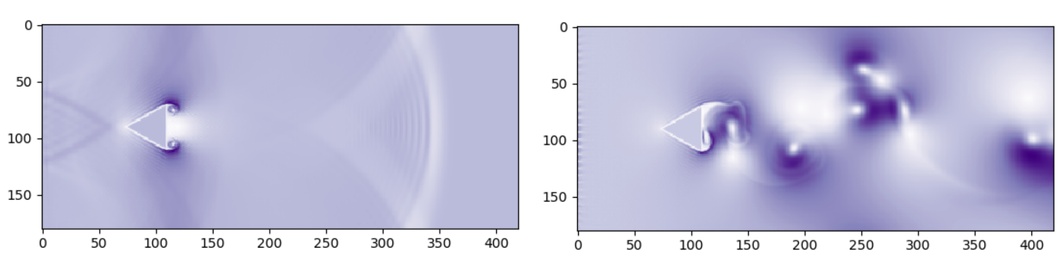

Figure 9: Obstacle: Triangle — Reynolds number: 1100

As observed from the above visualizations, we show the initial and final phases of our simulation and observe a proper fluid flow and particle interaction within our system. We can see, during the initial phases, a velocity wave propagation through our system, and the bouncing of these waves through the left and up down walls, which confirm that our boundary constraints are working well. Interaction of particle close to obstacle boundaries also confirm the same. We can observe, for low Reynolds number, our flow past the obstacle remains stable for longer times as compared to higher Reynolds numbers where we can start observing instabilities way faster, while also confirming the presence of Kármán vortex streets. The time and computation taken to simulate these systems, were relatively faster than and conceptually easier to code, making LBM method an efficient solver for such systems. The videos for these visualizations are available here [1].

4 Conclusions

4.1 Advantages of LBM:

- LBM offers a simpler and more intuitive approach to simulating fluid flows compared to traditional CFD methods, making it easier to implement and understand.
- The method is highly parallelizable and can efficiently utilize parallel computing resources, allowing for large-scale simulations on high-performance computing clusters.
- LBM can easily handle complex geometries, including porous media, moving boundaries, and multiphase flows, without the need for complex mesh generation. This flexibility makes LBM suitable for a wide range of applications.
- With appropriate numerical schemes and grid resolutions, LBM can achieve high accuracy in predicting fluid flow phenomena, making it a valuable tool for engineering and scientific simulations.

4.2 Limitations of LBM:

- LBM can struggle with numerical stability issues at very high Reynolds numbers, which are common in turbulent flows.And for flows with large gradients or stiff source terms, maintaining numerical stability without excessive damping can be problematic.
- LBM is inherently a compressible method, which means it can struggle to accurately simulate incompressible flows without additional modifications. The method typically requires low Mach number approximations to maintain accuracy in incompressible flow regimes.
- Since LBM models fluid flow at finite speeds, it inherently captures sound waves, which can be unnecessary for incompressible flow simulations and may require filtering.
- While LBM is grounded in kinetic theory, it lacks the robust theoretical foundation of some traditional CFD methods, such as the Navier-Stokes equations. This can sometimes make it harder to rigorously justify the method's accuracy and stability.
- Extending LBM to handle non-Newtonian fluids, which have complex, nonlinear stress-strain relationships, can be challenging and may require significant modifications to the basic algorithm.

In conclusion, LBM offers a versatile and efficient approach to simulating fluid flows across various disciplines. By simulating fluid flow at the microscopic level and leveraging simplified collision and propagation rules, LBM provides valuable insights into complex fluid dynamics phenomena, making it a valuable tool for researchers and engineers seeking to understand and analyze fluid flow behavior.

[1] https://github.com/the21vk/NM_2024_TermPapers/tree/main

References

[1] “Statistical Mechanics”, 2nd edition, K. Huang, Wiley.

[2] “Lees-Edwards boundary conditions for lattice Boltzmann”, A.J. Wagner and I. Pagonabarraga, J. Stat. Phys. 107, 521 (2002), [also cond-mat/0103218].

[3] Hohenberg and Halperin

[4] L. Zhang, S. Fent, P. Dong and S. Gao, “Lattice Boltzmann schemes for the nonlinear Schrodinger equation”, Phys. Rev. E 74, 036704 (2006)

[5] Mocz, P. (2022, November 29). Create your own Lattice Boltzmann simulation (With Python). Medium. https://medium.com/swlh/create-your-own-lattice-boltzmann-simulation-with-python-8759e8b53b1c

[6] Kármán vortex street. (2024, May 6). In Wikipedia. https://en.wikipedia.org/wiki/K