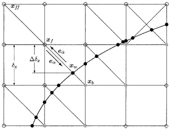
The Lattice Boltzmann Method

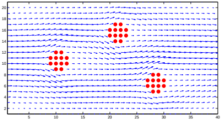
Computational Fluid Dynamics

Lindsay Crowl

GSAC Talk

November 7, 2006





How do you model fluid?

- Most complex fluid dynamics problems **cannot** be solved analytically
- We want to know how the fluid moves
- One way to describe the fluid's motion by its velocity profile

Typically we use the Incompressible **Navier Stokes equations** to solve for the flow dynamics.

Navier Stokes Equations

The vector $\mathbf{u}(\mathbf{x}, t)$ represents the fluid velocity at a space time point $(\mathbf{x}, t)$.

Conservation of mass equation:

$$\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0.$$

Momentum Equations:

$$\rho \left[\frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} + \mathbf{u}(\mathbf{x}, t) \cdot \nabla \mathbf{u}(\mathbf{x}, t) \right] = -\nabla p + \mu \nabla^2 \mathbf{u}(\mathbf{x}, t).$$

Navier Stokes Equations

ρ is the density of the fluid, ∇p is the pressure gradient, and μ is the fluid viscosity.

Conservation of mass equation:

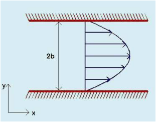
$$\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0.$$

Momentum Equations:

$$\rho \left[\frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} + \mathbf{u}(\mathbf{x}, t) \cdot \nabla \mathbf{u}(\mathbf{x}, t) \right] = -\nabla p + \mu \nabla^2 \mathbf{u}(\mathbf{x}, t).$$

Example: Using the Navier Stokes Equations

We can use the Incompressible Navier Stokes equations to solve a simple flow problem. Consider flow through a 2-dimensional pipe, driven by a pressure gradient in the x-direction.



$$\mathbf{u}(\mathbf{x}, t) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = \begin{bmatrix} u(x, y) \\ 0 \end{bmatrix}$$

$$p(\mathbf{x}, t) = p(x)$$

Example: Using the Navier Stokes Equations

In 2D, the conservation of mass equation,

$$\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0$$

can be written as

$$\frac{\partial u(x,y,t)}{\partial x} + \frac{\partial v(x,y,t)}{\partial y} = 0.$$

Since we know that

- $v(x,y,t) = 0$ (no flow in the vertical direction) and
- $u(x,y,t) = u(x,y)$,

the conservation equation is satisfied when $u(x,y)$ is independent of x.

Example: Using the Navier Stokes Equations

Since the pressure gradient is independent of time, so is the flow. Thus $\frac{\partial u}{\partial t} = 0$. The momentum equations

$$\rho \left[\frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} + \mathbf{u}(\mathbf{x}, t) \cdot \nabla \mathbf{u}(\mathbf{x}, t) \right] = -\nabla p + \mu \nabla^2 \mathbf{u}(\mathbf{x}, t)$$

simplifies to:

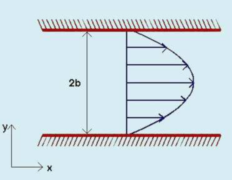
$$\nabla p(x) = \frac{\partial p(x)}{\partial x} = \mu \nabla^2 \mathbf{u} = \mu \frac{\partial^2 u(y)}{\partial y^2}.$$

Example: Using the Navier Stokes Equations

Given that the pressure is independent of y and the velocity is independent of x, we can argue that

$$\frac{\partial p(x)}{\partial x} = \mu \frac{\partial^2 u(y)}{\partial y^2} = C.$$

Thus, solving the quadratic in $u(y)$, we get parabolic flow!

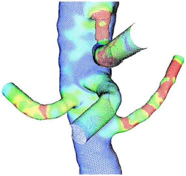


... Unfortunately most fluid dynamic problems aren't this simple.

Why do we care?

Biological Applications:

- Blood flow in large and small vessels
- Angiogenesis
- Clot Formation

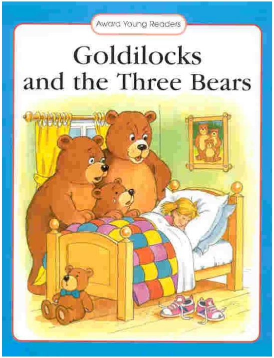


These are complex problems because

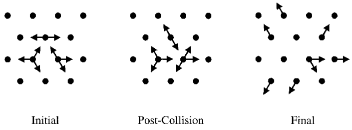
- Vessel boundaries are not perfect cylinders
- Blood is a non-Newtonian fluid.
- RBCs and platlets make it a collidal particle suspension

Modeling Methods

- Macroscopic
  - Continuum assumption
  - Navier Stokes equations
  - Scale is too large
- Microscopic
  - Models the molecular world
  - Computationally expensive
  - Scale is too small
- Mesoscopic
  - Lattice Boltzmann Method
  - Scale is juuuust right.



Origin of LBM: Lattice Gas Automata

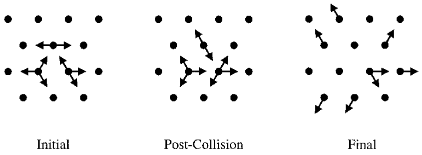


Particle collisions of the 2D-6 Velocity microscopic lattice-gas model.

- Particles exist on lattice grid
- Discrete Velocity field
- Exclusion Principle
- Symmetric Particle Collisions

It's so noisy! Quiet that stochastic stuff down! It's too loud!

Origin of LBM: Lattice Gas Automata



Indroduce the Boolean variable,

$$n_i(\mathbf{x}, t) = (1, 0) \text{ i.e. } (\text{yes/no}),$$

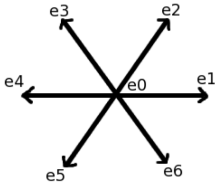
which describes whether or not there is a particle in a certain velocity direction ($\mathbf{e}_i$) at a space-time point ($\mathbf{x}$, t).

Deriving LBM from LGA

We can derive the Lattice Boltzmann Method from Lattice Gas Automata by determining the probability that there is a particle moving in the ith direction at $(\mathbf{x}, t)$:

$$f_i(\mathbf{x}, t) = \langle n_i(\mathbf{x}, t) \rangle \in [0, 1]$$

where the index "i" represents the velocity ($\mathbf{e}_i$) at a particular node.

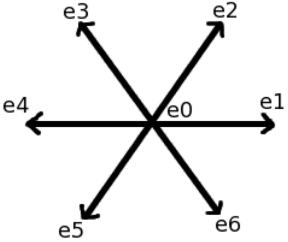


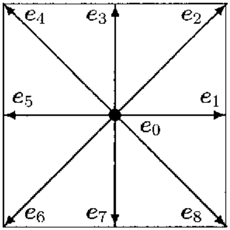
$f_i(\mathbf{x}, t)$ is the particle distribution function.

Deriving LBM from LGA

Typical LBM geometries (in 2-dimensions):

- Hexagonal: 7 Velocities
- Square: 9 Velocities





Alternative Derivation

LBM can also be derived by discretizing the Boltzmann Equation:

$$\underbrace{\frac{Df(\mathbf{x},\mathbf{e},t)}{Dt}}_{\text{Material Derivative}} = \underbrace{\frac{\partial f(\mathbf{x},\mathbf{e},t)}{\partial t} + \mathbf{e}\cdot\nabla f(\mathbf{x},\mathbf{e},t)}_{\text{Advection}} = \underbrace{\Omega(\mathbf{f}(\mathbf{x},\mathbf{e},\mathbf{t}))}_{\text{Collision}}$$

(derivation was proposed after LBM was created from LGA!)

Conservation Equations

The conserved macroscopic quantities of the Lattice Boltzmann Equation can be obtained by evaluating the hydrodynamic moments of $f(\mathbf{x}, t)$.

Fluid Density

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{e}, t) d\mathbf{e} \approx \sum_i f_i(\mathbf{x}, t)$$

Momentum Density

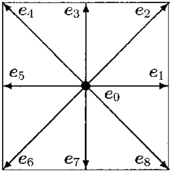
$$\rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{e}, t)\mathbf{e} d\mathbf{e} \approx \sum_i f_i(\mathbf{x}, t)\mathbf{e}_i$$

Momentum Flux

$$\Pi(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{e}, t)\mathbf{e}\mathbf{e}^T d\mathbf{e} \approx \sum_i f_i(\mathbf{x}, t)\mathbf{e}_i\mathbf{e}_i^T$$

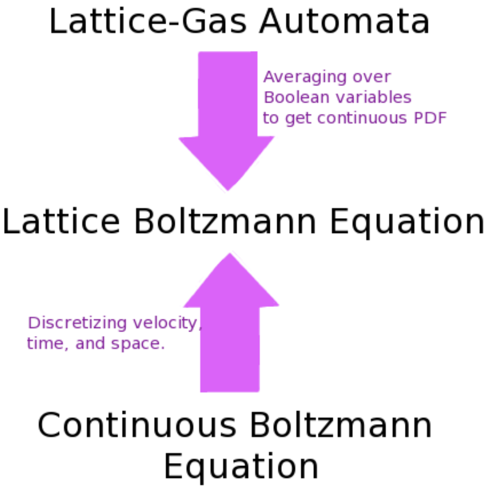
Stream and Collide Algorithm

- Particle distribution functions "live" on a lattice grid.
- Velocity is discretized, but no longer dealing with discrete particles.
- The PDFs $f_i(x,t)$ follows the LBM governing equation



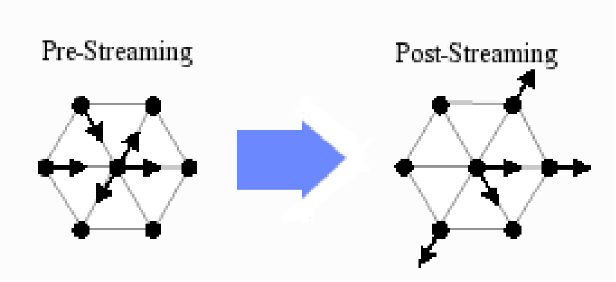
$$\underbrace{f_i(\mathbf{x}+\delta t\mathbf{e}_i, t+\delta t) - f_i(\mathbf{x},t)}_{\text{Advection}} = \underbrace{\Omega_i(f_0, f_1, f_2, \dots)}_{\text{Collision}}.$$

Derivation Recap



Streaming Step

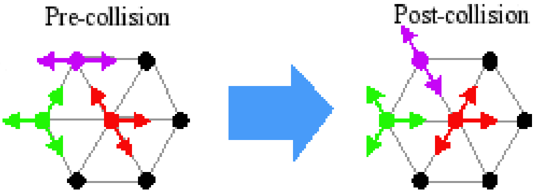
Move along the velocity direction to the next lattice grid node.



Collision Step

For the LGA model:

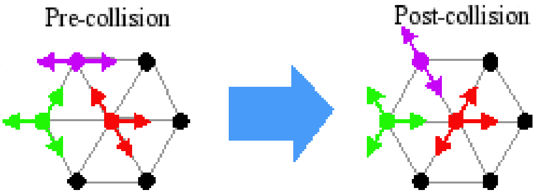
- Collisions must be symmetric
- Probability of various equivalent outcomes



Collision Step

What about for particle distributions in the LMB?

- Collision operator must also be symmetric
- There exists an equilibrium distribution, for which the collision operator does not change anything.



The Collision Operator

Given that $\Omega_i(\mathbf{f}^{eq}) = 0$ for all i, we can linearize about the equilibrium particle distribution,

$$\Omega_i(\mathbf{f}) \approx \Omega_i(\mathbf{f}^{eq}) + \frac{\partial \Omega_i}{\partial f_j}(f_j - f_j^{eq}) = \mathbf{L}_{ij}(f_j - f_j^{eq}),$$

where $\mathbf{L}_{ij}$ is the **collision matrix**.

The Collision Operator

The simpliest version of the collision matrix L_{ij} is

$$f_i(\mathbf{x} + \delta t \mathbf{e}_i, t + \delta t) - f_i(\mathbf{x}, t) = \frac{1}{\tau}\left(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)\right),$$

where τ is the relaxation time for all modes (determines fluid viscosity). This is called the **single-relaxation time** model. However, we can only use it in our algorithm if we know f_i^{eq}.

Equilibrium Distribution

The exact form of f_i^{eq} depends on lattice geometry. For 2D9V model it is:

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

where $c=\frac{dx}{dt}$, is the lattice grid spacing over the time step and w_i are the weights at each discrete velocity.

Need to find $\mathbf{u}(\mathbf{x},t)$ and $\rho(\mathbf{x},t)$ in order to use the equilibrium distribution in the linearized collision operator.

Conservation Equations

The conserved macroscopic quantities of the Lattice Boltzmann Equation can be obtained by evaluating the hydrodynamic moments of $f_i(\mathbf{x}, t)$.

Fluid Density

$$\rho(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t)$$

Momentum Density

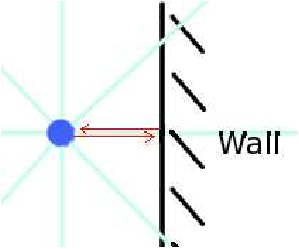
$$\rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t)\mathbf{e}_i$$

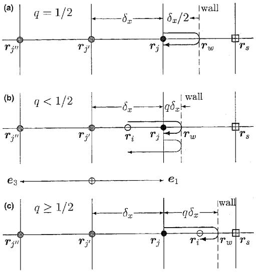
Algorithm Recap:

1. Streaming Step: move PDFs to next lattice node
2. Calculate the macroscopic parameters ρ and u by summing over lattice velocities at each node
3. Determine f_i^{eq} from ρ and u
4. Collision Step
5. Rinse and Repeat.

Bounce-back Boundary Conditions

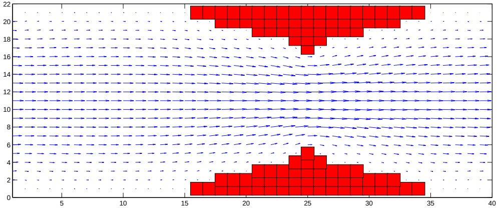
- What happens when the fluid is near a solid boundary?
- What if the boundary is curved?
- What if the boundary is also moving?





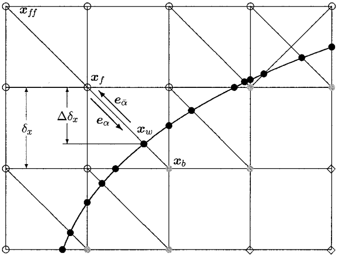
Simulation Results

Flow through a contracting vessel wall. Ugly "stair-stepping boundaries."



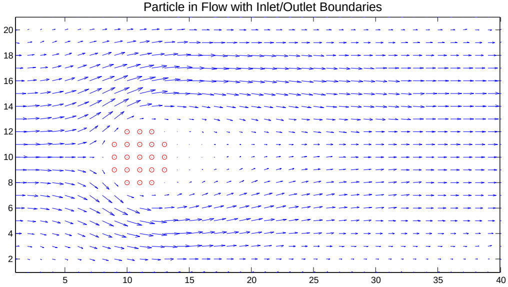
Complex Boundary Conditions

Curved boundaries, such as the boundary of a circular particle in the flow, complicate the bounce-back rule.



Simulation Results

Flow around a macroscopic particle in a 2D cylinder.
The top and bottom walls have no-slip (bounce-back) boundary conditions.
The left side is an inlet, and the right side is an outlet.



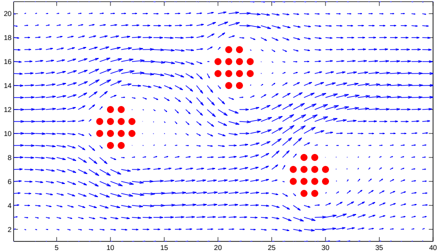
Moving Particles in Flow

In order to have moving particles in flow:

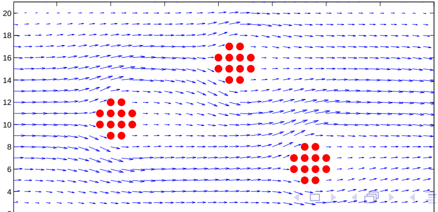
- Bounce-back BCs must be altered (momentum exchange)
- Particle velocity and angular velocity updated
- Particle's position in lattice changes as it moves with the flow

Simulation Results

Flow around stationary particles:



Flow around particles moving in flow:



Large Scale Simulation

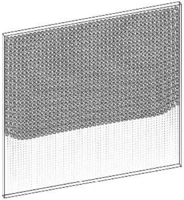


Fig. 22. Particle positions at $t = 2$ s.

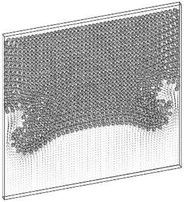


Fig. 23. Particle positions at $t = 4$ s.

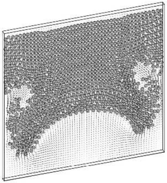


Fig. 24. Particle positions at $t = 5$ s.

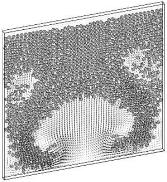


Fig. 25. Particle positions at $t = 6$ s.

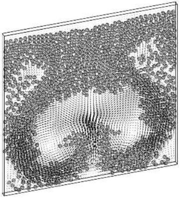


Fig. 26. Particle positions at $t = 7$ s.

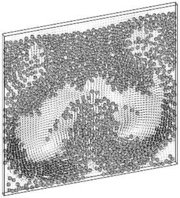


Fig. 27. Particle positions at $t = 8$ s.

Future Directions

- Model solid particles moving in the fluid
- Add external forces to flow
- Incorporate Immersed Boundary Method into LBM
  - Fluid Filled Particles
  - Deformable Boundary

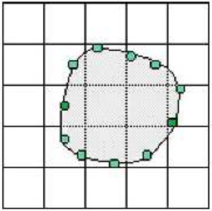


Fig. 1. A set of Lagangran boundary points for a two-dimensional particle.

Future Directions

- Model moving solid particles in flow
- Add external forces to flow
- Incorporate Immersed Boundary Method into LBM
  - Fluid Filled Particles
  - Deformable Boundary

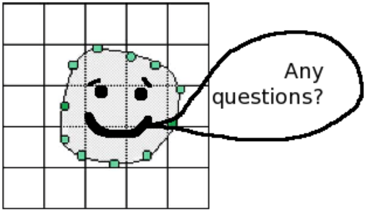


Fig. 1. A set of Lagangran boundary points for a two-dimensional particle.