CFD Lab

The Lattice-Boltzmann Method (LBM)

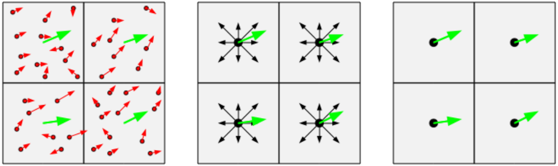
Nikola Tchipev

25.04.2014

Outline

LBM - a different story

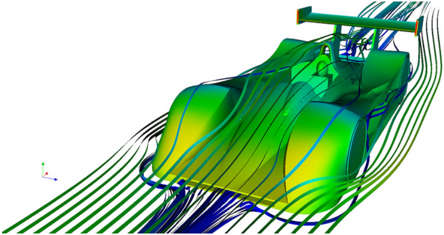
- Macroscale:
  - Finite Difference Methods (FDM)
- Mesoscale:
  - Lattice-Boltzmann Method (LBM)
- Microscale:
  - Molecular Dynamics (MD)



MD microscale | LBM mesoscale | FDM macroscale

LBM

- fluid solver, but we don’t solve NSE
- based on statistical mechanics
- new (and still evolving) method
- easy to program
- already a factor in the automotive industry



Some assumptions

What assumptions did we have in Worksheet 1?

Some assumptions

What assumptions did we have in Worksheet 1?

- inscompressible, isothermal, Newtonian, ...

Some assumptions

What assumptions did we have in Worksheet 1?

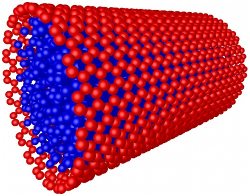
- inscompressible, isothermal, Newtonian, ...
- continuum assumption

Some assumptions

What assumptions did we have in Worksheet 1?

- inscompressible, isothermal, Newtonian, ...
- continuum assumption

Can we solve *any* flow problem with NSE?
Flow in a carbon nanotube?



The continuum assumption

Fluids in reality

composed of atoms and molecules, empty space in between

Fluids under the continuum assumption

composed of **continuous matter, filling the entire space**

When is the continuum assumption valid?

The continuum assumption

Continuum assumption is valid for

$$Kn << 1. \tag{1}$$

Kn: Knudsen number

$$Kn = \frac{\lambda}{L_c}, \tag{2}$$

L_c: characteristic length

λ: mean free path

- air at STP: $\lambda \approx \mathcal{O}(nm)$

Thought experiment

Small particle in fluid at rest

- $\Rightarrow \vec{u} = 0$ identically
- L_c - diameter of particle.
- as L_c decreases, Kn increases
- as Kn approaches 1, the particle begins to feel collisions with individual molecules
- Brownian motion kicks in!

But NSE (or the Stokes Eq.) predict no motion of the particle!

Outline

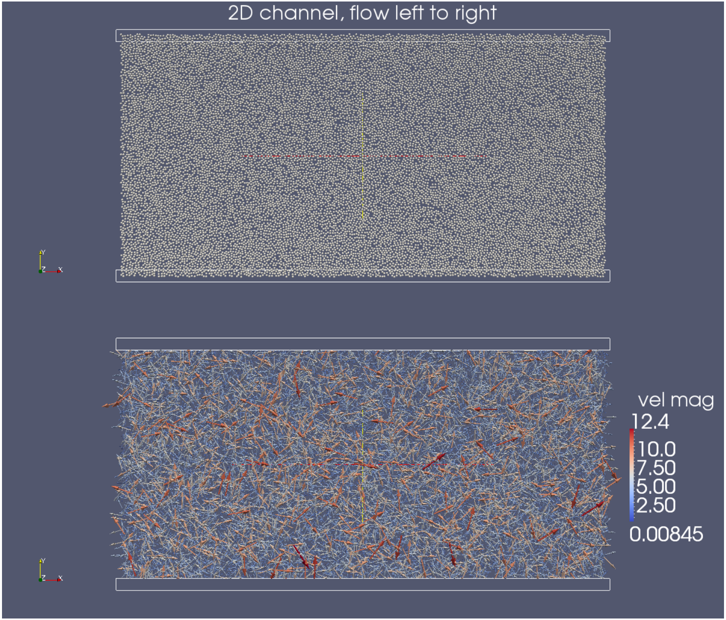
Molecular Dynamics

Nano-, Micro- things:

- Nano-, Microfluidics

Applications

- nanotubes, -pores, -filters...
- Lab-on-a-chip
- pop-up touch screens?


2D channel, flow left to right
vel mag
12.4
10.0
7.50
5.00
2.50
0.00845

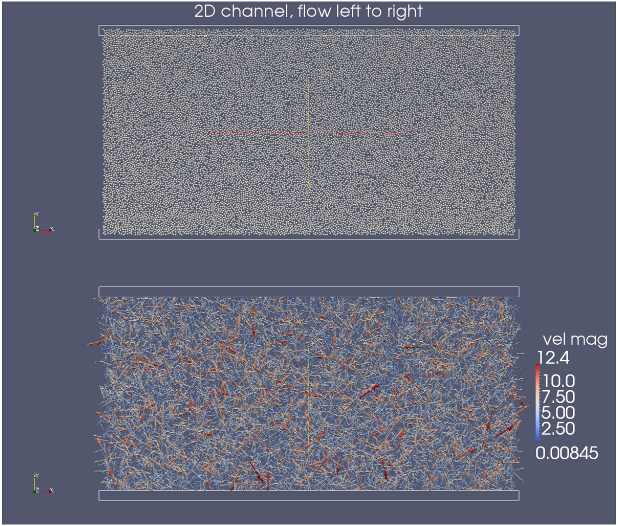
Can we solve any flow problem with MD?

Largest MD simulation - 4×10^{12} particles.
1 mililiter of water - 3×10^{22} particles

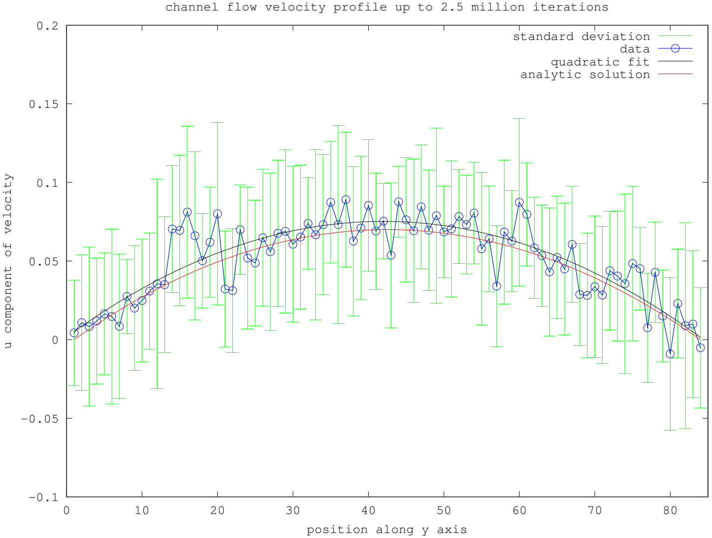
Statistical noise

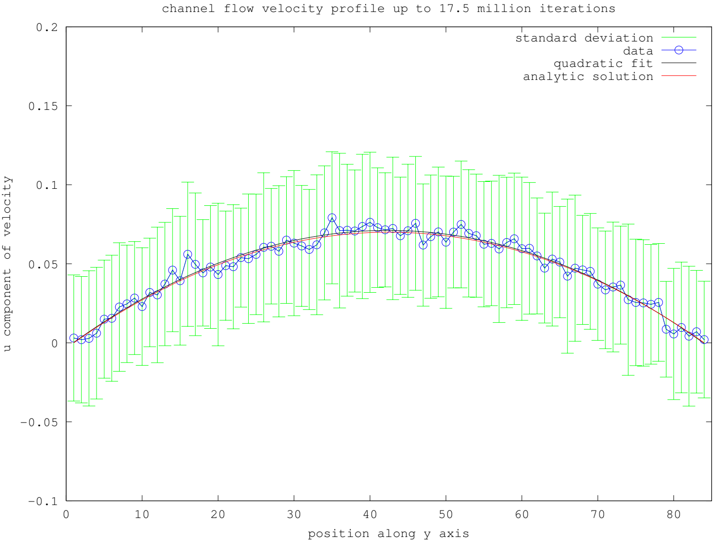
Need to sample:

- in space
- in time
- Monte-Carlo-type



mean speed: 0.17, max speed: 12.4


channel flow velocity profile up to 2.5 million iterations
standard deviation
data
quadratic fit
analytic solution
u component of velocity
position along y axis
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
10
20
30
40
50
60
70
80


channel flow velocity profile up to 17.5 million iterations
standard deviation
data
quadratic fit
analytic solution
u component of velocity
position along y axis
0.2
0.15
0.1
0.05
0
−0.05
−0.1
0
10
20
30
40
50
60
70
80

Outline

Lattice Gas Cellular Automata

LGCA

- reduce MD computational complexity
- replace molecule-to-molecule force calculations by rigid body collisions

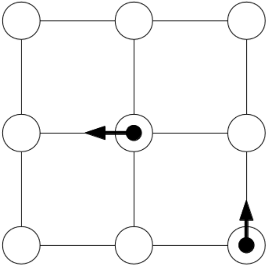
Cellular Automaton

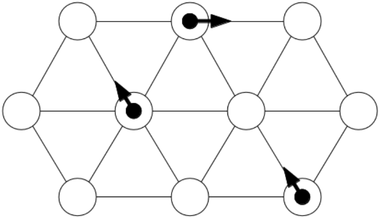
- Conway's Game of Life (CGL)
- regular grid of cells
- each cell can have a discrete number of states
(alive or dead in CGL)

The Lattice in LGCA and LBM

lattice $\approx$ graph

- at the end of a timestep, particles can reside only at vertices
- during a timestep, particles can travel only along one edge





The Lattice in LGCA and LBM

Discretize space and velocity

Macro-view

no big deal, we always discretize

Micro-view

very big deal! Only one of several velocities $\{\vec{c}_i\}_{i=0,\dots,n}$ allowed.
This restricts:

- velocity direction
- velocity magnitude
- information transfer (LGCA and LBM are fully explicit schemes)

Note a lattice may include $\vec{c}_i = 0$: stationary particles.

LGCA

Exclusion principle

no two particles can have the same position AND velocity at the same time

State of a cell

a few boolean variables:

- $n_i(\vec{r}, t) = 1$
  if we have a particle at position $\vec{r}$, with velocity $\vec{c}_i$, at time t
- $n_i(\vec{r}, t) = 0$
  otherwise

LGCA

Algorithm

while (t != t_{end}) $\{$

1. collide - handle multiple particles present at the same site
2. stream - travel the respective edge
3. $t = t + \Delta t$

$\}$

Stream

$$n_i^{in}(\vec{r} + \vec{c}_i \Delta t, t + \Delta t) = n_i^{out}(\vec{r}, t) \tag{3}$$

Collide

$$n_i^{out}(\vec{r}, t) = n_i^{in}(\vec{r}, t) + \Omega_i(n_i^{in}(\vec{r}, t)) \tag{4}$$

Ω: collision operator

Rigid body collisions:

- conserve mass (number of particles):

$$\sum_i n_i^{in} = \sum_i n_i^{out} \tag{5}$$

- conserve momentum

$$\sum_i n_i^{in} \vec{c}_i = \sum_i n_i^{out} \vec{c}_i \tag{6}$$

Square lattice collision

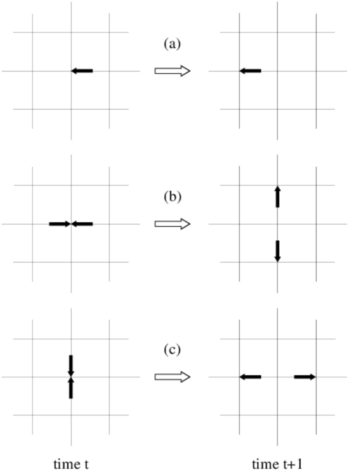


image: [1]

Hexagonal lattice collision

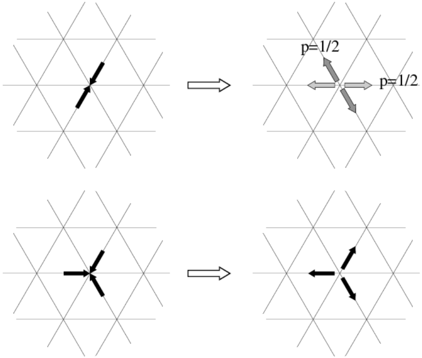


image: [1]

- probabilistic
- look-up table

Some boundary conditions

image: [1]

Some boundary conditions

image: [1]

No-slip in LGCA

- very easy
- very convenient

Some boundary conditions

image: [1]

No-slip in LGCA

- very easy
- very convenient

regular grid + easy No-slip BC

- very well suited for handling complex geometries
- no Meshing!

A flow problem

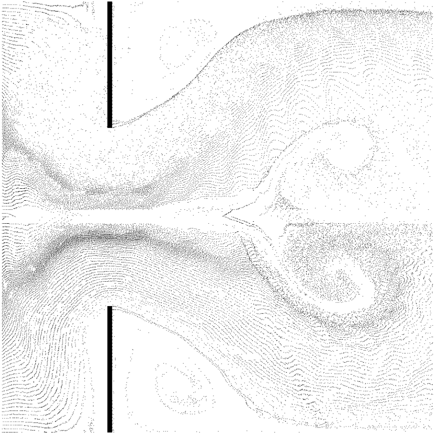


image: [1]

LGCA

NSE can be mathematically recovered from LGCA!

Recap

- very fast and simple compared to MD
- perfect for flow in complex geometries
- easy to parallelize
- statistical noise?

Outline

LBM

LGCA + statistics (statistical mechanics) + additional assumptions

Replace boolean n_i with real f_i.

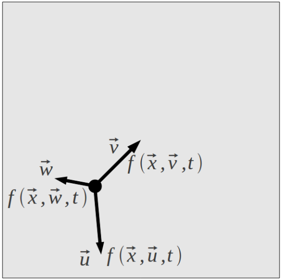
Algorithm

while (t != t_{end}) $\{$

1. collide - handle f_i's at the same site
2. stream - travel the respective edge
3. $t = t + \Delta t$

$\}$

The f_i



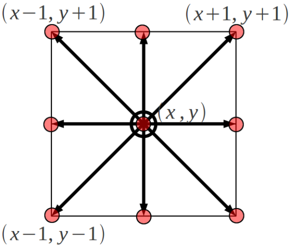
$f(\vec{x}, \vec{v}, t)$: probability density function for finding particles with velocity $\vec{v}$ at $(\vec{x}, t)$

$f \in \mathbb{R}, f \in (0, 1)$

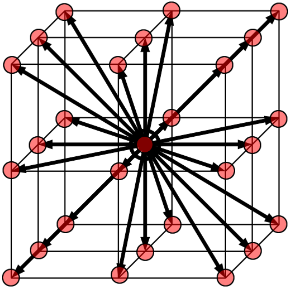
The LBM lattices

DnQm notation:

- n: number of dimensions
- m: number of directions



D2Q9



D3Q27

other possibilities: D2Q5, D2Q7, D3Q15, D3Q19

Macroscopic quantities

Given $\{f_i\}$, compute $\{\rho, u, p\}$:

- density:

$$\rho(\vec{x}, t) = \sum_{i=0}^{Q-1} f_i \approx 1$$

- momentum:

$$\vec{u}(\vec{x}, t)\rho(\vec{x}, t) = \sum_{i=0}^{Q-1} f_i \cdot c_i$$

- pressure:

$$p = \rho \cdot c_s^2$$

$c_s = \frac{1}{\sqrt{3}}$: speed of sound
local operations

Given $\{\rho, u\}$, can we compute $\{f_i\}$?

The equilibrium distribution

$f_i^{eq}(\rho, u)$

a specific mapping from $\{\rho, u\}$ to $\{f_i\}$:

$$f_i^{eq} = w_i \rho \left(1 + \frac{\vec{c}_i \cdot \vec{u}}{c_s^2} + \frac{1}{2c_s^4} (\vec{c}_i \cdot \vec{u})^2 - \frac{1}{2c_s^2} \vec{u} \cdot \vec{u} \right),$$

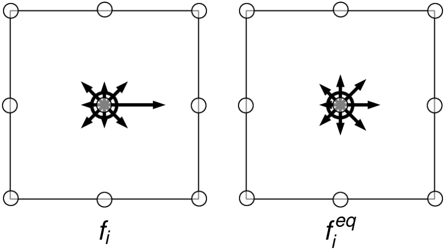
w_i - weights, depending on the chosen lattice.
E.g. D2Q9:

$$w_i = \begin{cases} \frac{4}{9} & \text{if } \|c_i\| = 0 \\ \frac{1}{9} & \text{if } \|c_i\| = 1 \\ \frac{1}{36} & \text{if } \|c_i\| = \sqrt{2} \end{cases}$$

- needed for IC, BC and collisions

The equilibrium distribution

$f^{eq} \approx$ Normal distribution



The equilibrium distribution

f^{eq} - Maxwell Boltzmann equilibrium distribution

$$f^{eq}(\vec{x}, \vec{c}_i, t) = \left(\frac{m}{2\pi k_B T}\right)^{D/2} \cdot \exp\left(-\frac{m(\vec{c}_i - \vec{u}(\vec{x}, t))^2}{2k_B T}\right),$$

it **assumes** an ideal gas:

$$pV = Nk_b T$$

$$p = \frac{N}{V} \times const = \rho \times const$$

Moreover, the form

$$f_i^{eq} = w_i \rho \left(1 + \frac{\vec{c}_i \cdot \vec{u}}{c_s^2} + \frac{1}{2c_s^4} (\vec{c}_i \cdot \vec{u})^2 - \frac{1}{2c_s^2} \vec{u} \cdot \vec{u} \right),$$

assumes $u << c_i$.

So, we assume a **low Mach number**:

$$Ma = \frac{u}{c_s} << 1$$

c_s - speed of sound, information transfer

$$c_s = const \times \sqrt{k_b T}$$

weakly compressible flow

Collision

BGK approximation

Bhatnagar-Gross-Krook approximation

$$\Omega_i = -\frac{1}{\tau}(f_i - f_i^{eq}(\rho, u))$$

$\tau \in (0.5, 2.0)$ - relaxation “time”

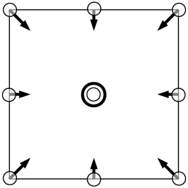
$$\tau = \frac{\nu}{c_s^2} + 0.5$$

ν - kinematic viscosity

we model “weak departure from equilibrium state of an ideal gas”

Collision II

Why collision is local

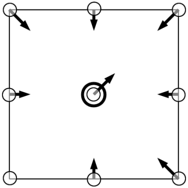


“pull” scheme

Distributions “pass through” each other.

Collision II

Why collision is local

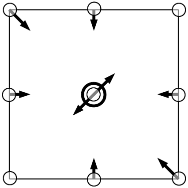


“pull” scheme

Distributions “pass through” each other.

Collision II

Why collision is local

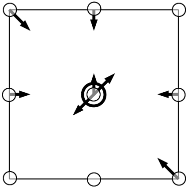


“pull” scheme

Distributions “pass through” each other.

Collision II

Why collision is local

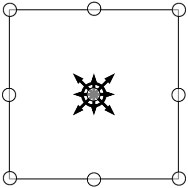


“pull” scheme

Distributions “pass through” each other.

Collision II

Why collision is local



"pull" scheme

Distributions "pass through" each other.

Collision III

Collision with intuition

$$\text{collision:} \qquad f_i := f_i + \Omega_i$$
$$f_i := f_i - \frac{1}{\tau}\left(f_i - f_i^{eq}\right)$$

Let $\omega = \frac{1}{\tau} \in (0.5, 2.0)$:

$$f_i := (1-\omega)f_i + \omega f_i^{eq},$$

Collision III

Collision with intuition

$$\text{collision:} \qquad f_i := f_i + \Omega_i$$
$$f_i := f_i - \frac{1}{\tau}\left(f_i - f_i^{eq}\right)$$

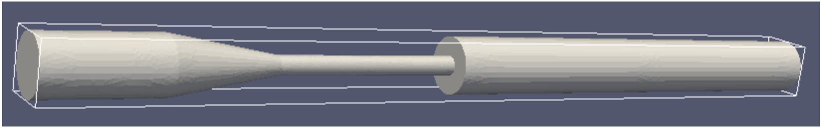
Let $\omega = \frac{1}{\tau} \in (0.5, 2.0)$:

$$f_i := (1-\omega) f_i + \omega f_i^{eq},$$

SOR update rule?

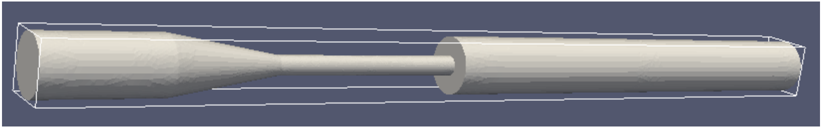
Collision III

Channel with a narrowing, flow from left to right:

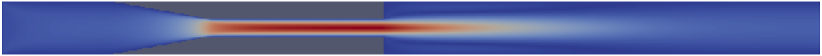


Collision III

Channel with a narrowing, flow from left to right:

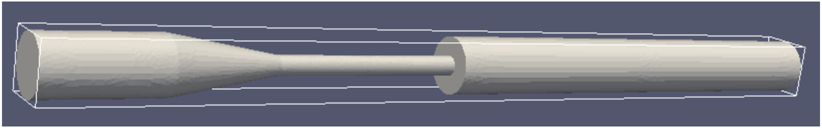


Velocity magnitude at high *Re*:

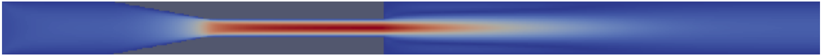


Collision III

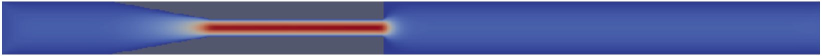
Channel with a narrowing, flow from left to right:

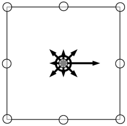


Velocity magnitude at high *Re*:

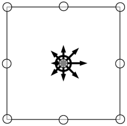


.. at low *Re*:





$(1-\omega)f_i$



ωf_i^{eq}

$$\omega = \omega(\nu) = \omega(Re)$$

Collision and streaming

Combined update rule

$$f_i\left(\vec{x} + c_i \Delta t, t + \Delta t\right) = f_i\left(\vec{x}, t\right) - \frac{1}{\tau}\left(f_i\left(\vec{x}, t\right) - f_i^{eq}\left(\rho\left(\vec{x}, t\right), u\left(\vec{x}, t\right)\right)\right)$$

Implementation

collide:

$$f_i^*\left(\vec{x}, t\right) := f_i\left(\vec{x}, t\right) - \frac{1}{\tau}\left(f_i\left(\vec{x}, t\right) - f_i^{eq}\left(\rho\left(\vec{x}, t\right), u\left(\vec{x}, t\right)\right)\right)$$

stream:

$$f_i\left(\vec{x} + c_i \Delta t, t + \Delta t\right) = f_i^*\left(\vec{x}, t\right)$$

NSE can be recovered!

multiscale Chapman Enskog analysis

The continuous case

We are solving the continuous Boltzmann equation

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \nabla = \Omega$$

with our particular Lattice space- and velocity discretization.

- density:

$$\rho(\vec{x}, t) = \int_{\mathbb{R}^D} f(\vec{v}) d\vec{v}$$

- momentum:

$$\vec{u}(\vec{x}, t)\rho(\vec{x}, t) = \int_{\mathbb{R}^D} f(\vec{v}) \cdot \vec{v} d\vec{v}$$

Stability

We made many assumptions ...

- $\rho \approx 1$
- $u << 1$
- ν can't get arbitrarily small
  $\Rightarrow$ how to control Re?

Pros

- can handle continuum problems (e.g. cars), but also higher *Kn* numbers
- coupling to MD
- closer to true physical description than NSE:
  turbulence, diffusion, multi-component flows

Cons

- Homework: compare with NSE

Outline

References

[1] Bastien Chopard, Introduction to Lattice Gas Cellular Automata,
`theory.physics.unige.ch/cours/outils/fichiers/lga.ps`