Lattice Boltzmann Methods

David Chas Bolton

Brandon Schwartz

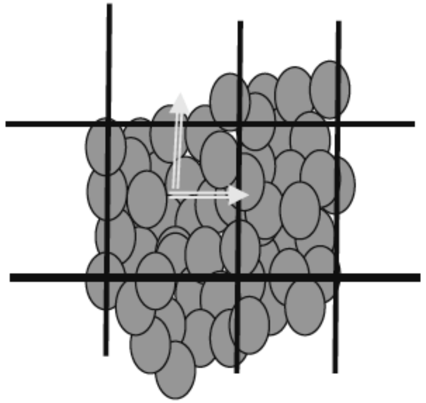
Srisharan Shreedharan

Introduction

Introduction

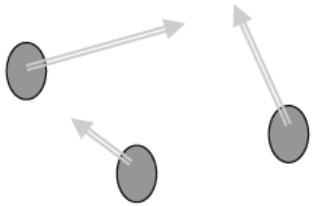
Continuum(Macroscopic scale), finite difference, finite volume, finite element, etc),

Navier-Stokes Equations



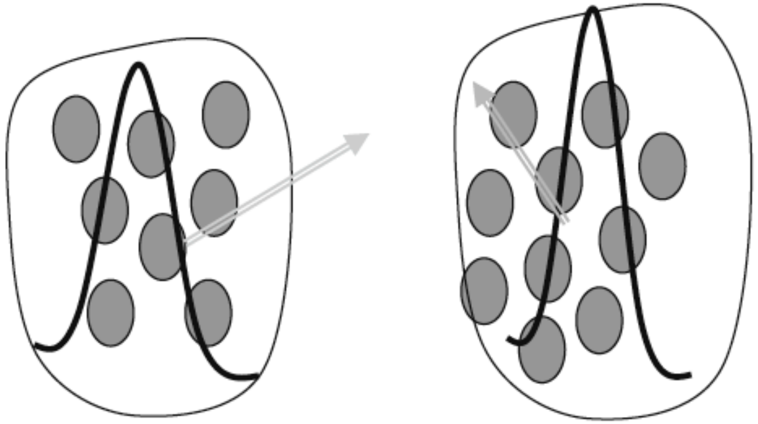
Molecular Dynamics (Microscopic scale),

Hamilton's Equation.



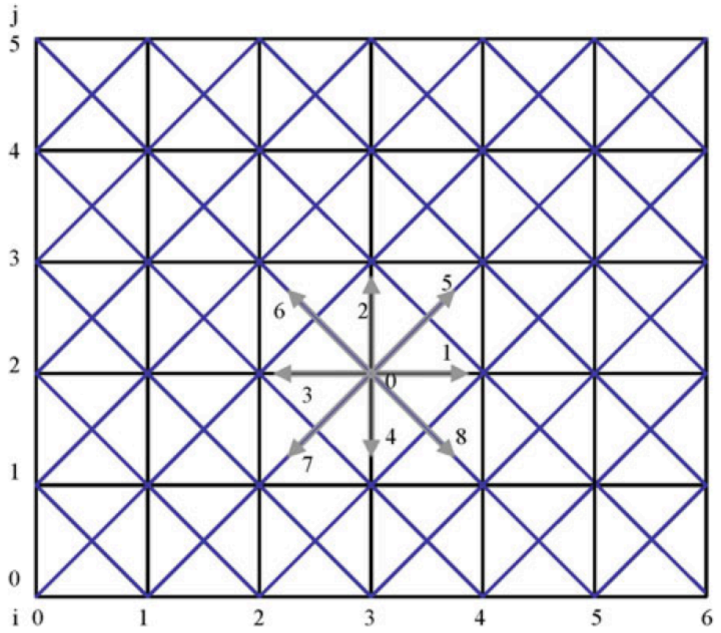
Lattice Boltzmann Method (Mesoscopic scale),

Boltzmann Equation



Introduction

- Fictitious assemblage of molecules
- Uses particle probability distribution function instead of simulating every molecule’s position and velocity
- Particles can only move from node to node within a lattice or between lattices, based on prescribed boundary conditions.
- Incompressible flow is assumed and particles ‘stream’ & ‘collide’



Introduction

LBM & FEM:

- Lattice <--> Mesh
- Boltzmann equation <——> Navier-Stokes equation
- Weighting parameter <——> Interpolation function

LBM & DEM

- Mesoscopic parameters are used to estimate macroscale properties (density, velocity, internal energy)

Introduction

LBM vs conventional CFD:

- Uses 1st order advection PDE instead of 2nd order convection PDE
- Discretization is implicit in Boltzmann equation
- Solved as a ’stream’ step and ‘collision’ step over all lattices and simple kinetic boundaries applied

Introduction

LBM advantages:

- Supports massive parallel computing since local lattice-level steps can be solved independently and simultaneously
- No need of ‘interface’ elements for multi-component/multi-phase fluid flows
- Multi-scale studies over wide range of particle sizes possible

LBM drawbacks:

- Needs more memory/storage than Navier-Stokes solvers
- Cannot stably handle compressible flows or Mach numbers higher than 0.3
- Requires external packages for THM coupling

Historical perspective

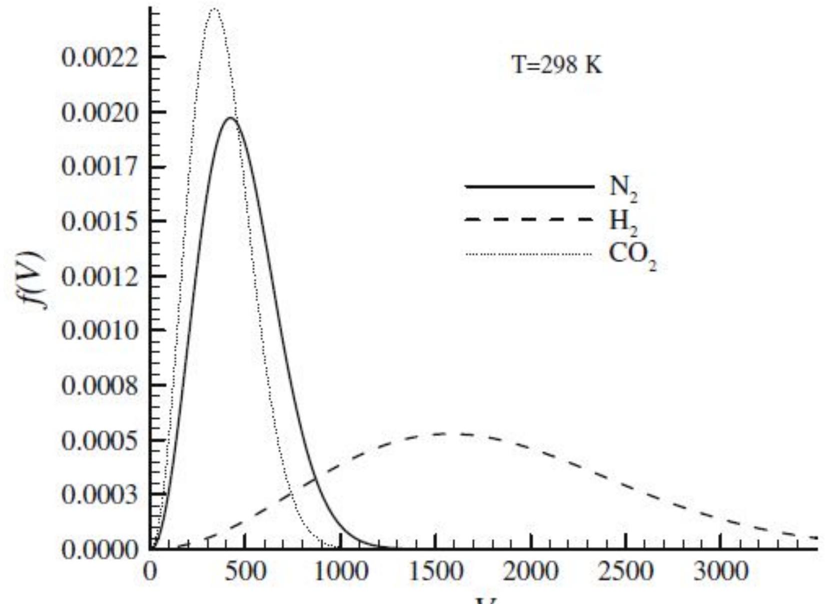
Historical Perspective

- LBM formulated in 1988 by McNamara and Zanetti
  - 1859: Maxwell’s distribution function
  - 1868: Boltzmann transport equation
  - 1954: Bhatnagar, Gross, and Krook (BGK) collision operator
  - 1956: FEM by Turner
  - 1973,76: Hardy, Pomeau, and de Pazzis (HPP) model/Lattice Gas Automata (LGA)
  - 1980: Finite volume method (FVM) at Imperial College

Maxwell Distribution Function

$$\iiint f(c_x)f(c_y)f(c_z)\,\mathrm{d}c_x\,\mathrm{d}c_y\,\mathrm{d}c_z = 1.$$

$$\langle c^2 \rangle = \int_0^\infty c^2 f(c)\,\mathrm{d}c = \frac{3kT}{m}$$



- Measures the probability that a certain percentage of a population of molecules will be traveling at a certain speed
- Heavier molecules travel slower (on average)
- The area under each distribution is 1

Boltzmann Transport Equation/BGK Collision Operator

- If no collisions

$$f(r + c\mathrm{d}t, c + F\mathrm{d}t, t + \mathrm{d}t)\mathrm{d}r\mathrm{d}c - f(r, c, t)\mathrm{d}r\mathrm{d}c = 0$$

- Same equation, with collisions

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial r} \cdot c + \frac{F}{m} \cdot \frac{\partial f}{\partial c} = \Omega$$

- If no external force

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

- BGK Collision Operator

$$\Omega = \omega(f^{\mathrm{eq}} - f) = \frac{1}{\tau}(f^{\mathrm{eq}} - f)$$

- LBM Equation:

$$\frac{\partial f_i}{\partial t} + c_i \nabla f_i = \frac{1}{\tau}(f_i^{\mathrm{eq}} - f_i)$$

Discretized LBM Equation

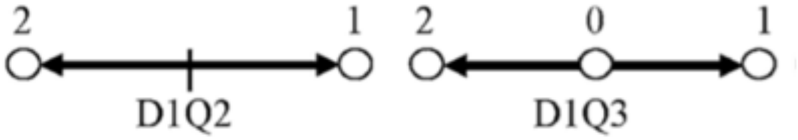
$$f_i(r + c_i\Delta t, t + \Delta t) = f_i(r,t) + \frac{\Delta t}{\tau}[f_i^{\mathrm{eq}}(r,t) - f_i(r,t)]$$

- Turns 1st order PDE into algebraic expression
- Addresses challenges previous CFM’s did not
- Very straightforward to use

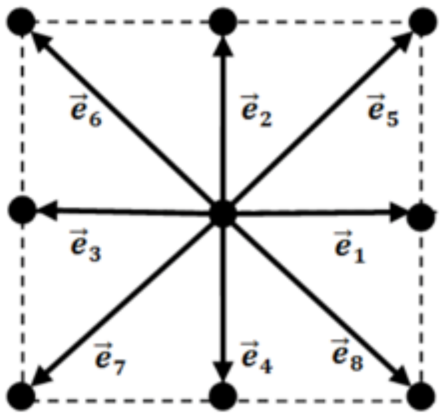
General principles & equations

Lattice Arrangements

- Description of the lattice and degree of problem is represented via D_nQ_m
- m=speed, # of the linkages of a node, number of velocity directions
- N= dimension of the problem
- Particles are restricted to move via linkages and are allowed to interact at nodes
- Particles move along the linkages at the lattice speed; normally assume that in a given time step the particles move from one cell node to the next.



Example of a 1d problem
Source: A.A Mohamad

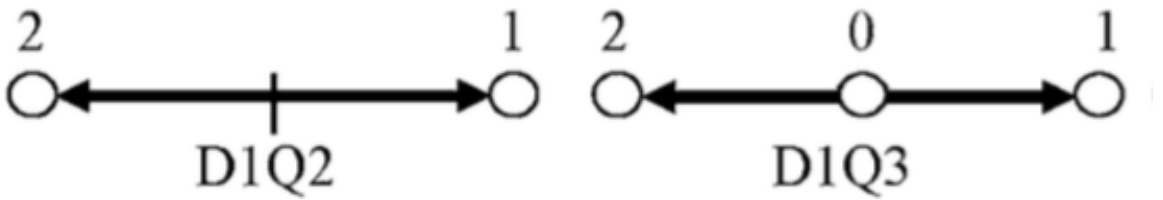


D_2Q_9

Example of a 2d problem
Source:http://www.cims.nyu.edu/~billbao/report930.pdf

Lattice Arrangements

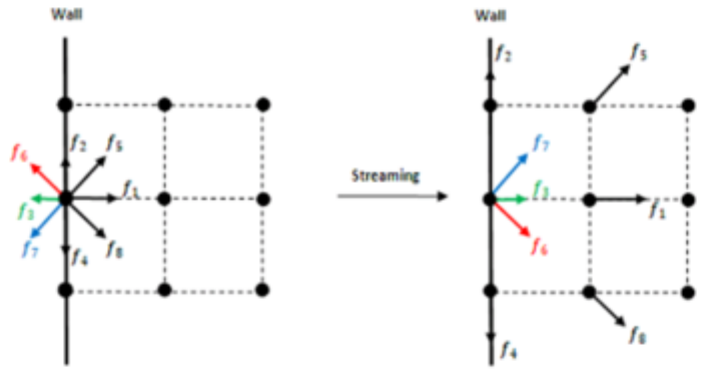
- D1Q3 is described with three velocities c0,c1,c2 and f0,f1f2. c0=0 for center particle
- Total number of particles not allowed to exceed 3
- Particle's are free to move to the left or right
- Each particle is assigned a particular weight, which is a function of how close that particle is to the central node and the velocities.
- For D1Q3 the weighting factors, $\boldsymbol{\omega}_l$, are 4/6, 1/6, 1/6 for f0,f1,f2
- Speed of sound, Cs, is 1/(3^.5)
- The sum of all weights must equal 1.

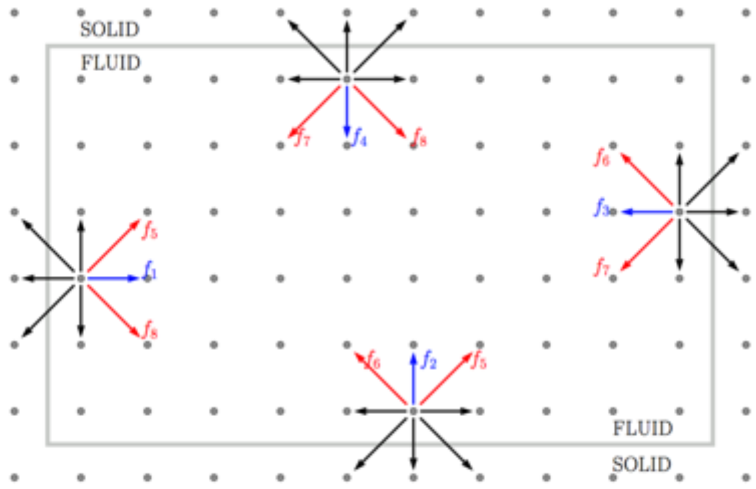


Boundary Conditions

Bounce-Back:

- Models solid stationary or moving boundary conditions.
- When a particle encounters the boundary it will simply bounce back.
- Boundary can be placed between the nodes or going through the center of the nodes.
- Unknown distributions after collision are f2,f5 and f6.
- Focusing on bottom layer we see that f2=f4, f6=f8, f5=f7.



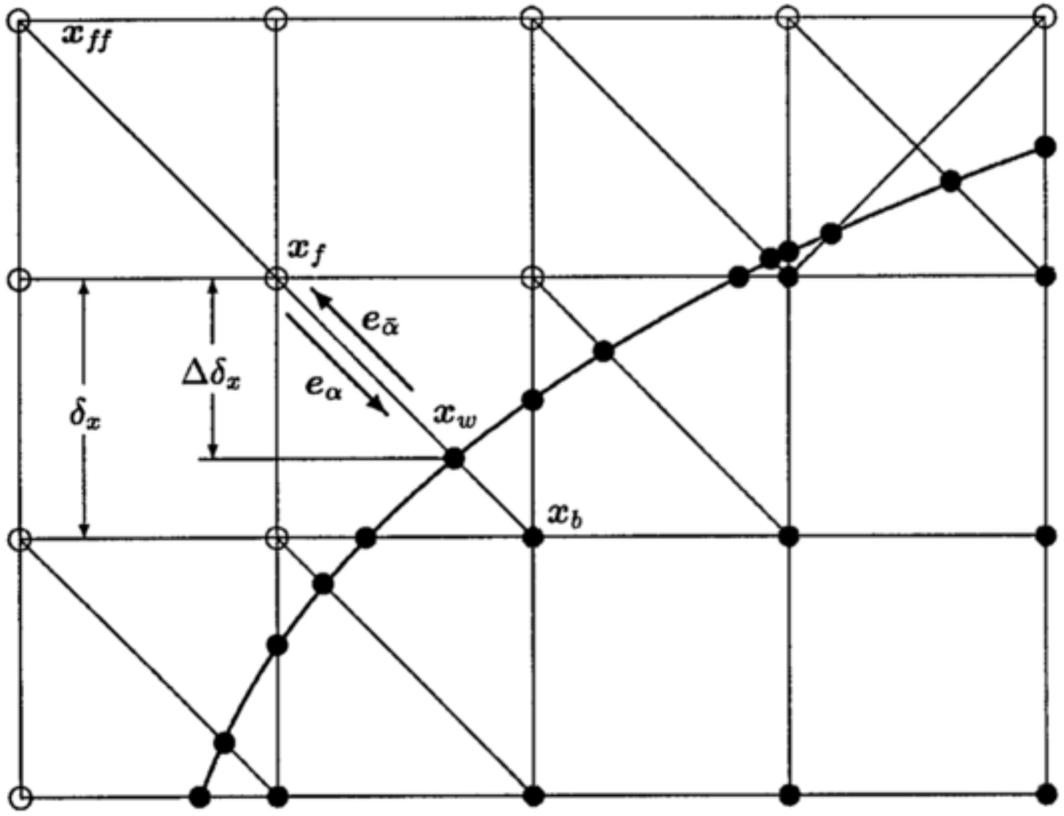


Igor 2013

Curved Boundary Conditions

Represent the curved surface through a set of stair steps.

Requires the boundary to placed between the nodes.



Mei et al. 2000

From Lattice Gas Automata to LBM

- For LGA particles restricted to move within a lattice
- We represent the particles in space and time via $n_i(\mathbf{x}, t), \quad i = 0, ...M.$
- X=position, t=time and i=direction of the particle velocity
- Ni= 1=> particle is present at site x and time t vice versa if Ni=0
- Can describe how the particles evolve in space and time via:

$$n_i(\mathbf{x} + \mathbf{e}_i \delta t, t + 1) = n_i(\mathbf{x}, t) + \Omega_i(n(\mathbf{x}, t)), \quad i = 0, ...M$$

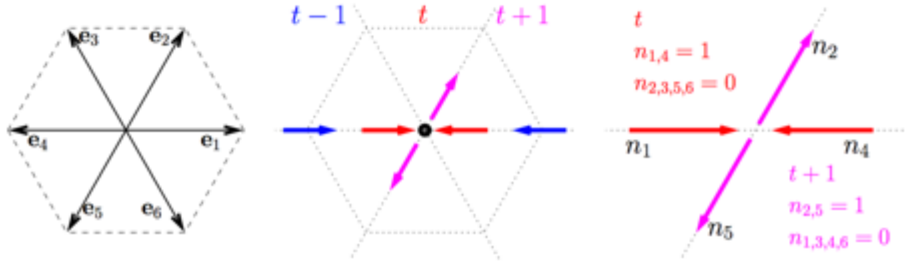
- e_i= local particle velocities, $\boldsymbol{\Omega}_i$=collision operator
- Collisions are local

Example of LGA

At time t-1 particle is occupied at site 1 and 4

At time t, particles collide

At time t+1, particles move off in directions of e2 and e5. (governed by scattering rules)



Derivation of Lattice Boltzmann Equation from LGA

- Rather than describing particles via Boolean algebra we can represent them through a distribution function
- Fk=average (nk)
- Distribution function, f(x,e,t); where x=position, e=velocity, t=time
- If we apply some force,f, on the particles their positions and velocities will change from x $\rightarrow$ x+edt; e $\rightarrow$ e +F/ Mdt

Collison vs no Collison

- If no collisions between particles take place, then the distribution of particles should be the same before and after force was applied i.e

$$f(\mathbf{x} + \mathbf{e}dt, \mathbf{e} + \frac{\mathbf{F}}{m}dt, t + dt)d\mathbf{x}d\mathbf{e} = f(\mathbf{x}, \mathbf{e}, t)d\mathbf{x}d\mathbf{e}.$$

- With collisions there will be a difference between initial distribution and final distribution:

$$f(\mathbf{x} + \mathbf{e}dt, \mathbf{e} + \frac{\mathbf{F}}{m}dt, t + dt)d\mathbf{x}d\mathbf{e} - f(\mathbf{x}, \mathbf{e}, t)d\mathbf{x}d\mathbf{e} = \Omega(f)d\mathbf{x}d\mathbf{e}dt.$$

- Divide through by dxdedt $\Rightarrow$ $\frac{\mathcal{D}f}{dt} = \Omega(f).$

- Where $\boldsymbol{\Omega}$(f) is the collision operator

Lattice Boltzmann Equation final form

- Rate of change of our distribution function is equal to the collision operator

$$\frac{\mathcal{D}f}{dt} = \Omega(f).$$

- Expanded form:

$$\mathcal{D}f = \frac{\partial f}{\partial \mathbf{x}}d\mathbf{x} + \frac{\partial f}{\partial \mathbf{e}}d\mathbf{e} + \frac{\partial f}{\partial t}dt,$$

- Divide through by dt:

$$\frac{\mathcal{D}f}{dt} = \frac{\partial f}{\partial \mathbf{x}}\frac{d\mathbf{x}}{dt} + \frac{\partial f}{\partial \mathbf{e}}\frac{d\mathbf{e}}{dt} + \frac{\partial f}{\partial t}$$

- Note, e=dx/dt; de/dt=F/m
- If we assume F=0, i.e no external forces then:

$$\frac{\partial f}{\partial t} + \mathbf{e}\cdot\nabla f = \Omega(f).$$

Collision Operator continued

- If particles in our system collide, then it must take some time for the particles to reach an equilibrium state.
- The time taken to reach the equilibrium state is a function of the type of collision and a relaxation time
- Due to the complexity of the Collision Operator the Boltzmann equation can be difficult to solve.
- We can solve for the collision operator based on Bhatnagar, Gross and Krook solution

More on the collision operator

The collision operator $\boldsymbol{\Omega}$(f) is replaced with the BGK operator: $\Omega_k = -\frac{1}{\tau}\left(f_k - f_k^{\mathrm{EQ}}\right)$

τ= is the relaxation rate towards equilibrium and is related to viscosity by : $\nu = \frac{2\tau - 1}{6}\frac{(\Delta x)^2}{\Delta t}$

should be in the range of .5-2

fk^{EQ}= equilibrium distribution function

fk^{EQ} is an expansion of the Maxwell Distribution Function assuming a low Mach number: M=u/cs <<1

Where u=macroscopic velocity of the fluid, cs=speed of sound, $\boldsymbol{\rho}$=macroscopic fluid velocity

$$f = \frac{\rho}{2\pi/3} e^{-\frac{3}{2}(\mathbf{e}-\mathbf{u})^2} = \frac{\rho}{2\pi/3} e^{-\frac{3}{2}(\mathbf{e}\cdot\mathbf{e})} e^{\frac{3}{2}(2\mathbf{e}\cdot\mathbf{u}-\mathbf{u}\cdot\mathbf{u})};$$

Equilibrium Distribution Function, fk^{EQ}

- Note, Taylors Expansion for e^-x= 1-x +x^2/2-x^3/3!
- Using Taylors Expansion we can rewrite the equilibrium distribution function as follows:

$$f = \frac{\rho}{2\pi/3} e^{-\frac{3}{2}(\mathbf{e}\cdot\mathbf{e})} \left[1 + 3(\mathbf{e}\cdot\mathbf{u}) - \frac{3}{2}(\mathbf{u}\cdot\mathbf{u}) + \frac{9}{2}(\mathbf{e}\cdot\mathbf{u})^2\right]$$

$$f_k^{\mathrm{EQ}} = \rho w_k \left[1 + 3(\mathbf{e}_k\cdot\mathbf{u}) - \frac{3}{2}(\mathbf{u}\cdot\mathbf{u}) + \frac{9}{2}(\mathbf{e}_k\cdot\mathbf{u})^2\right]$$

- k=number of velocities, $\boldsymbol{\omega}_k$ = weighting factors

Going from continuous form to discretized

Recall, that the collision operator, $\boldsymbol{\Omega}$(f), is the rate of change of particle distribution function.

$$\frac{\mathcal{D}f}{dt} = \Omega(f).$$

Expanding the particle distribution function out into its counterparts, we obtain the equation to the right:

$$\mathcal{D}f = \frac{\partial f}{\partial \mathbf{x}} d\mathbf{x} + \frac{\partial f}{\partial \mathbf{e}} d\mathbf{e} + \frac{\partial f}{\partial t} dt,$$

Again, dividing through by dt, and assuming no external forces yields the following:

$$\frac{\partial f}{\partial t} + \mathbf{e} \cdot \nabla f = \Omega(f).$$

Continuous to discrete

- Recall, that the collision operator is simply: $\boldsymbol{\Omega}$(f)= -1/τ(f-f^{eq})
- -1/τ(f-f^{eq})= ∂f/ ∂t+ ∂f/ ∂x *c
- Now multiply through by dt
- -dt/ τ(f-f^{eq})= (∂f/ ∂t + ∂f/ ∂x *c)dt (1)
- Note, Taylor series expansion: f(x + Δx, t + Δt)= f(x,t) + Δf + c * (Δf/ Δx) Δt
- Substitute the second term in the Taylor Series with Eq 1
- f(x + Δx, t + Δt)= f(x,t) - Δt/ τ (f-f^{eq}) = Discretized version of LBM

Connecting microscopic quantities to macroscopic quantities

- Basic idea: To relate microscopic phenomena to macroscopic behavior
- We can represent the density of a fluid via the following eq:

$$\rho(r,t) = \int m f(r,c,t)\,\mathrm{d}c$$

- Can represent the fluid velocity via the following eq:

$$\rho(r,t)u(r,t) = \int m c f(r,c,t)\,\mathrm{d}c$$

- Kinetic Energy:

$$\rho(r,t)e(r,t) = \frac{1}{2}\int m u_a^2 f(r,c,t)\,\mathrm{d}c$$

Hand calculation

Hand Calculation

- Imagine a long tube of gas with an initial temperature of $T = 0$.
- At times greater than 0, the left boundary of the tube has a temperature $T = 1$.
- Model the change in gas temperature throughout the tube as time increases
  - Assume the tube is non-conductive such that all heat transfer occurs through the gas

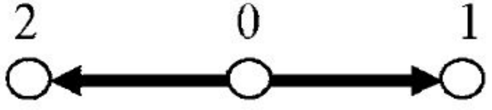
Problem Description

- Can be modeled at 1-D problem:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

$$\alpha = \frac{k}{\rho C}$$

$$\alpha = 0.25$$



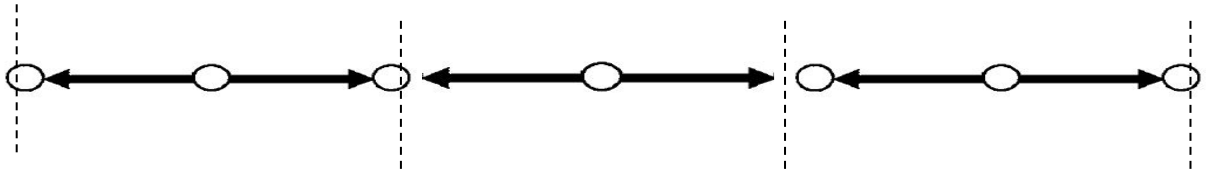
D1Q3

$$c_0 = 0, c_1 = 1, c_2 = -1$$

$$w_0 = \frac{4}{6}, w_1 = \frac{1}{6}, w_2 = \frac{1}{6}$$

$$\Delta x = 1, \Delta t = 1$$

- We will model with 3 elements:



Workflow

1. Initialize macroscopic properties and distribution functions
    1. T_w=1, all others 0.
    2. Make an educated guess for distribution function (for diffusion equation, it doesn't really matter)
        1. For initial fi's, set fi's in element 1 to wi's and fi's in elements 2 and 3 to ci's.
2. Calculate equilibrium distribution functions

$$f_0(1,0) = \frac{4}{6}, f_1(1,0) = \frac{1}{6}, f_2(1,0) = \frac{1}{6} \rightarrow T(1,0) = f_0 + f_1 + f_2 = 1$$

$$f_0(2,0) = 0, f_1(2,0) = 1, f_2(2,0) = -1 \rightarrow T(2,0) = 0$$

$$f_0(3,0) = 0, f_1(3,0) = 1, f_2(3,0) = -1 \rightarrow T(3,0) = 0$$

$$f_i^{eq}(x,t) = w_i T(x,t)$$

$$f_0^{eq}(1,0) = \frac{4}{6}, f_1^{eq}(1,0) = \frac{1}{6}, f_2^{eq}(1,0) = \frac{1}{6}$$

$$f_i^{eq}(2,0) = f_i^{eq}(3,0) = 0$$

After Initialization…

3. Calculate Collisions:

$$f_i^*(x,t) = (1-\omega)f_i(x,t) + \omega f_i^{eq}(x,t)$$

    1. Using the BGK Approximation for the Collision Operator

4. Calculate Streaming: $f_i(x + c_i\Delta t, t + \Delta t) = f_i^*(x,t)$

$$f_0(x, t + \Delta t) = f_0^*(x,t)$$
$$f_1(x + \Delta x, t + \Delta t) = f_1^*(x,t)$$
$$f_2(x - \Delta x, t + \Delta t) = f_2^*(x,t)$$

$$\mathit{Collision:}\ f_i^*(x,t) = (1-\omega) f_i(x,t) + \omega f_i^{eq}(x,t)$$

$$\alpha = \tau - \frac{\Delta t}{2}$$

$$f_0^*(1,0) = \left(1 - \frac{4}{3}\right)\left(\frac{4}{6}\right) + \frac{4}{3}\left(\frac{4}{6}\right) = \frac{4}{6}$$
$$f_1^*(1,0) = \left(1 - \frac{4}{3}\right)\left(\frac{1}{6}\right) + \frac{4}{3}\left(\frac{1}{6}\right) = \frac{1}{6}$$
$$f_2^*(1,0) = \left(1 - \frac{4}{3}\right)\left(\frac{1}{6}\right) + \frac{4}{3}\left(\frac{1}{6}\right) = \frac{1}{6}$$

$$\omega = \frac{\Delta t}{\tau}$$

$$0.25 = \frac{1}{\omega} - \frac{1}{2} \rightarrow \omega = \frac{4}{3}$$

$$f_0^*(2,0) = \left(1 - \frac{4}{3}\right)(0) + \frac{4}{3}(0) = 0$$
$$f_1^*(2,0) = \left(1 - \frac{4}{3}\right)(1) + \frac{4}{3}(0) = -\frac{1}{3}$$
$$f_2^*(2,0) = \left(1 - \frac{4}{3}\right)(-1) + \frac{4}{3}(0) = \frac{1}{3}$$

$$f_0^*(3,0) = \left(1 - \frac{4}{3}\right)(0) + \frac{4}{3}(0) = 0$$
$$f_1^*(3,0) = \left(1 - \frac{4}{3}\right)(1) + \frac{4}{3}(0) = -\frac{1}{3}$$
$$f_2^*(3,0) = \left(1 - \frac{4}{3}\right)(-1) + \frac{4}{3}(0) = \frac{1}{3}$$

$$Streaming: f_i(x + c_i\Delta t, t + \Delta t) = f_i^*(x, t)$$

$$f_0(x, t + \Delta t) = f_0^*(x, t)$$
$$f_1(x + \Delta x, t + \Delta t) = f_1^*(x, t)$$
$$f_2(x - \Delta x, t + \Delta t) = f_2^*(x, t)$$

$$f_0(1,1) = f_0^*(1,0) = \frac{4}{6}$$
$$f_0(2,1) = f_0^*(2,0) = 0$$
$$f_0(3,1) = f_0^*(3,0) = 0$$

$$f_1(3,1) = f_1^*(2,0) = -\frac{1}{3}$$

$$f_1(2,1) = f_1^*(1,0) = \frac{1}{6}$$

$$f_1(1,1) = f_1^*(1,0) = \frac{1}{6}\ (B.C.)$$

$$f_2(1,1) = f_2(1,0) = \frac{1}{6}(B.C.)$$

$$f_2(2,1) = f_2^*(3,0) = \frac{1}{3}$$

$$f_2(3,1) = f_2^*(3,0) = \frac{1}{3}$$

Update Macroscopic Properties

$$T(x,t) = \sum_{i=1}^{3} f_i(x,t)$$

$$T(1,1) = f_0(1,1) + f_1(1,1) + f_2(1,1) = \frac{4}{6} + \frac{1}{6} + \frac{1}{6} = 1$$

$$T(2,1) = f_0(2,1) + f_1(2,1) + f_2(2,1) = 0 + \frac{1}{6} + \frac{1}{3} = \frac{1}{2}$$

$$T(3,1) = f_0(3,1) + f_1(3,1) + f_2(3,1) = 0 + \left(-\frac{1}{3}\right) + \frac{1}{3} = 0$$

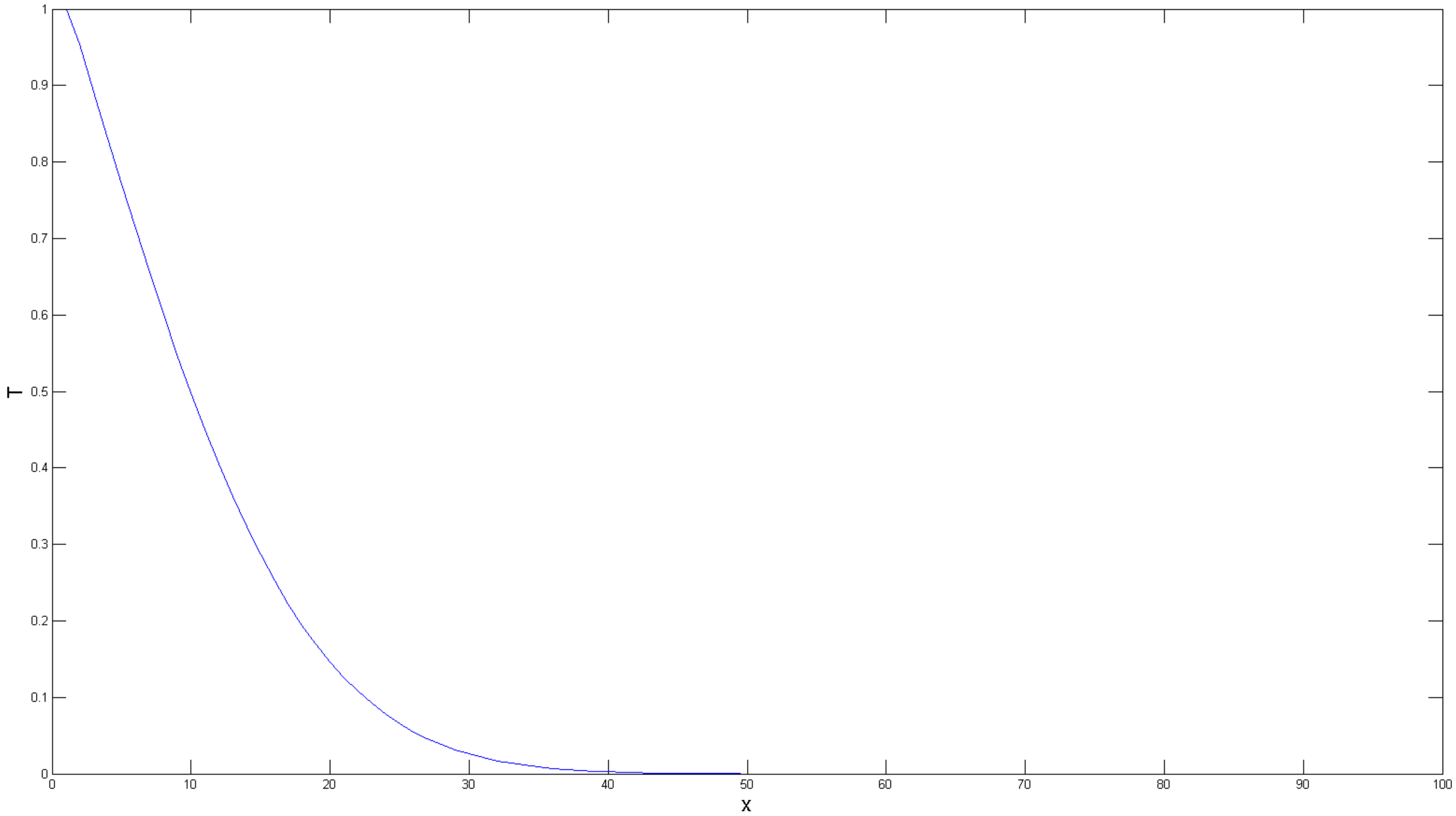
Update Macroscopic Properties

$$f_i^{eq}(x,t) = w_i T(x,t)$$
$$f_0^{eq}(1,1) = \frac{4}{6}(1) = \frac{4}{6}$$
$$f_1^{eq}(1,1) = \frac{1}{6}(1) = \frac{1}{6}$$
$$f_2^{eq}(1,1) = \frac{1}{6}(1) = \frac{1}{6}$$

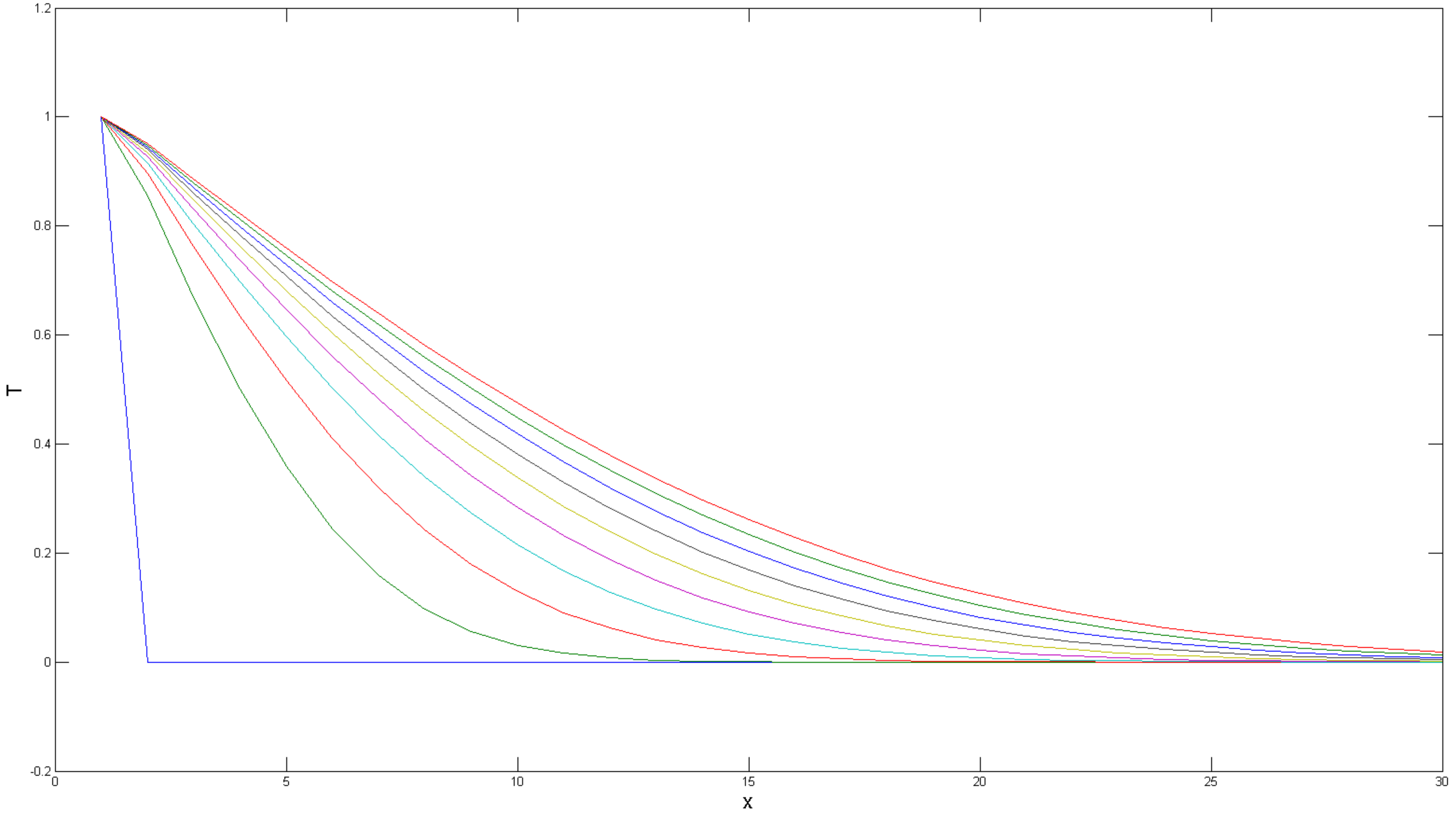
$$f_0^{eq}(2,1) = \frac{4}{6}\left(\frac{1}{2}\right) = \frac{1}{3}$$
$$f_1^{eq}(2,1) = \frac{1}{6}\left(\frac{1}{2}\right) = \frac{1}{12}$$
$$f_2^{eq}(2,1) = \frac{1}{6}\left(\frac{1}{2}\right) = \frac{1}{12}$$

$$f_0^{eq}(3,1) = \frac{4}{6}(0) = 0$$
$$f_1^{eq}(3,1) = \frac{1}{6}(0) = 0$$
$$f_2^{eq}(3,1) = \frac{1}{6}(0) = 0$$

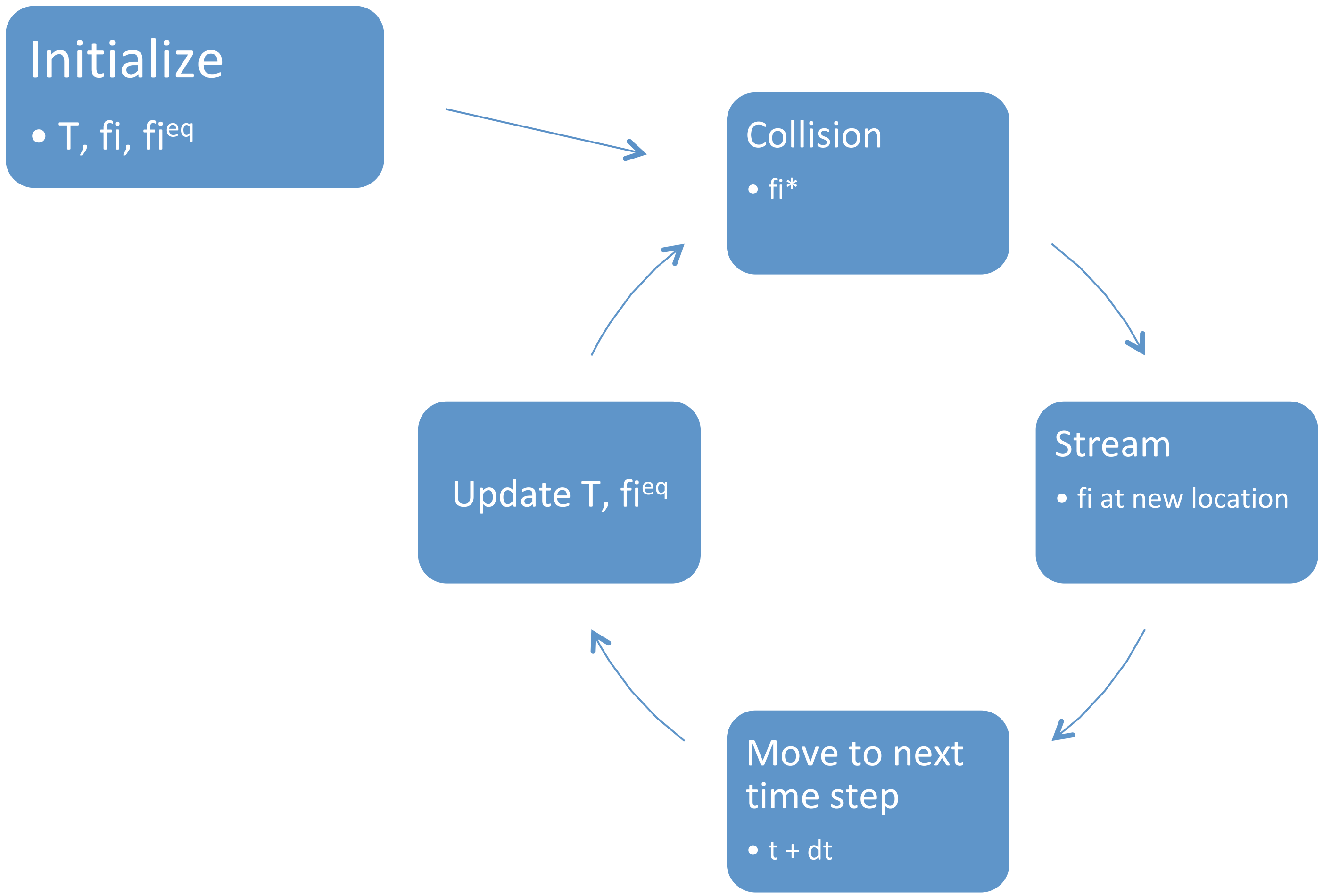
Same problem for 100 units



Form of the solution with increasing T



Summary


Initialize
• T, fi, fi^eq
Collision
• fi*
Stream
• fi at new location
Move to next time step
• t + dt
Update T, fi^eq

Numerical example using OpenLB

Example problem – 2D flow around cylinder

- Steady flow around a cylinder in a channel
- Poiseuille flow profile at inlet
- Dirichlet boundary of p=0 at outlet
- Elastic bounce-back along walls
- Reynolds number = 20 and 100 for laminar and turbulent flows respectively
- D2Q9 system

2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	5	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	5	5	0	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	5	0	0	0	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	5	5	0	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	5	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

0=Do nothing
1=Fluid
2=no slip/bounce back boundary

3=velocity boundary
4=constant (zero in our case) boundary
5=curved boundary (cylinder)

cylinder2d — cylinder2d — 115×40

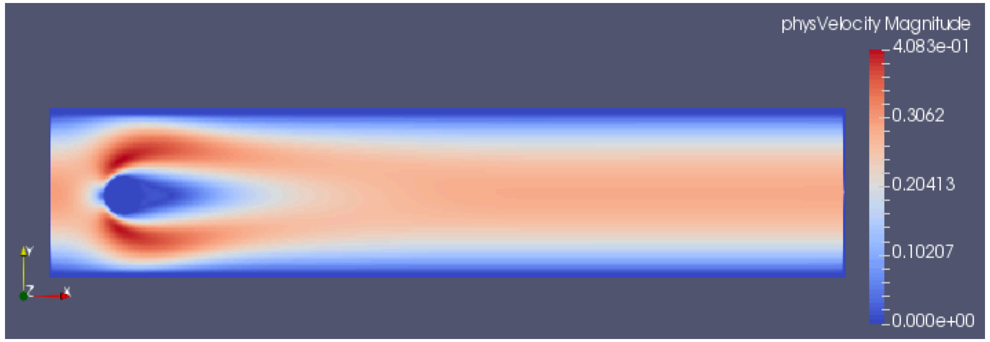
```
-bash: cd: cylinder2d: Not a directory
Srisharans-MBP:cylinder2d srisharan$ ./cylinder2d
[LBconverter] LBconverter information
[LBconverter] characteristical values
[LBconverter] Dimension(d):                     dim=2
[LBconverter] Characteristical length(m):       charL=0.1
[LBconverter] Characteristical speed(m/s):      charU=0.2
[LBconverter] Characteristical time(s):         charT=0.5
[LBconverter] Density factor(kg/m^d):           charRho=1
[LBconverter] Characterestical mass(kg):        charMass=0.01
[LBconverter] Characterestical force(N):        charForce=0.004
[LBconverter] Characterestical pressure(Pa):    charPressure=0.04
[LBconverter] Pressure level(Pa):               pressureLevel=0
[LBconverter] Phys. kinematic viscosity(m^2/s): charNu=0.0002
[LBconverter] lattice values
[LBconverter] DeltaX:                           deltaX=0.05
[LBconverter] Lattice velocity:                 latticeU=0.02
[LBconverter] DeltaT:                           deltaT=0.001
[LBconverter] Reynolds number:                  Re=100
[LBconverter] DimlessNu:                        dNu=0.01
[LBconverter] Viscosity for computation:        latticeNu=0.004
[LBconverter] Relaxation time:                  tau=0.512
[LBconverter] Relaxation frequency:             omega=1.95312
[LBconverter] conversion factors
[LBconverter] latticeL(m):                      latticeL=0.005
[LBconverter] Time step (s):                    physTime=0.0005
[LBconverter] Velocity factor(m/s):             physVelocity=10
[LBconverter] FlowRate factor(m^d/s):           physFlowRate=0.05
[LBconverter] Mass factor(kg):                  physMass=2.5e-05
[LBconverter] Force factor(N):                  physForce=0.5
[LBconverter] Force factor massless(N/kg):      physMasslessForce=20000
[LBconverter] Pressure factor(Pa):              physPressure=400
[LBconverter] latticePressure:                  latticeP=0.01
[prepareGeometry] Prepare Geometry ...
[SuperGeometryStatistics2D] updated
[SuperGeometryStatistics2D] updated
[SuperGeometry2D] cleaned 238 outer boundary voxel(s)
[SuperGeometry2D] the model is correct!
[CuboidGeometry2D] ---Cuboid Stucture Statistics---
[CuboidGeometry2D]  Number of Cuboids:  7
```

```
[LatticeStatistics] step=29800; t=14.9; uMax=0.0430039; avEnergy=0.000260975; avRho=1.00005
[getResults] pressure1=0.0808003; pressure2=-0.0149993; pressureDrop=0.0957996; drag=3.03107; lift=0.233117
[Timer] step=30000; percent=93.75; passedTime=65.355; remTime=4.357; MLUPs=14.2383
[LatticeStatistics] step=30000; t=15; uMax=0.04319; avEnergy=0.000261459; avRho=1.00007
[getResults] pressure1=0.0814775; pressure2=-0.0144718; pressureDrop=0.0959493; drag=3.03612; lift=0.313639
[Timer] step=30200; percent=94.375; passedTime=65.732; remTime=3.9178; MLUPs=19.5777
[LatticeStatistics] step=30200; t=15.1; uMax=0.0433338; avEnergy=0.00026197; avRho=1.00007
[getResults] pressure1=0.0815696; pressure2=-0.0145817; pressureDrop=0.0961513; drag=3.04155; lift=0.350273
[Timer] step=30400; percent=95; passedTime=66.112; remTime=3.47958; MLUPs=19.3716
[LatticeStatistics] step=30400; t=15.2; uMax=0.0434153; avEnergy=0.000262455; avRho=1.00006
[getResults] pressure1=0.0811046; pressure2=-0.0152404; pressureDrop=0.096345; drag=3.04652; lift=0.334963
[Timer] step=30600; percent=95.625; passedTime=66.625; remTime=3.0482; MLUPs=14.3493
[LatticeStatistics] step=30600; t=15.3; uMax=0.0434215; avEnergy=0.000262861; avRho=1.00004
[getResults] pressure1=0.0801173; pressure2=-0.016321; pressureDrop=0.0964383; drag=3.04916; lift=0.267066
[Timer] step=30800; percent=96.25; passedTime=66.996; remTime=2.61023; MLUPs=19.8951
[LatticeStatistics] step=30800; t=15.4; uMax=0.0433488; avEnergy=0.000263148; avRho=1.00001
[getResults] pressure1=0.0787531; pressure2=-0.0176492; pressureDrop=0.0964023; drag=3.04939; lift=0.153858
[Timer] step=31000; percent=96.875; passedTime=67.359; remTime=2.17287; MLUPs=20.2231
[LatticeStatistics] step=31000; t=15.5; uMax=0.0432043; avEnergy=0.000263296; avRho=0.999988
[getResults] pressure1=0.0772504; pressure2=-0.0190148; pressureDrop=0.0962652; drag=3.04788; lift=0.00987331
[Timer] step=31200; percent=97.5; passedTime=67.885; remTime=1.74064; MLUPs=14.0213
[LatticeStatistics] step=31200; t=15.6; uMax=0.0430067; avEnergy=0.000263313; avRho=0.999964
[getResults] pressure1=0.0759254; pressure2=-0.0202007; pressureDrop=0.0961262; drag=3.04677; lift=-0.144876
[Timer] step=31400; percent=98.125; passedTime=68.253; remTime=1.3042; MLUPs=19.9491
[LatticeStatistics] step=31400; t=15.7; uMax=0.0428223; avEnergy=0.000263233; avRho=0.999947
[getResults] pressure1=0.0750589; pressure2=-0.0210431; pressureDrop=0.096102; drag=3.0484; lift=-0.287684
[Timer] step=31600; percent=98.75; passedTime=68.633; remTime=0.868772; MLUPs=19.3716
[LatticeStatistics] step=31600; t=15.8; uMax=0.043045; avEnergy=0.000263106; avRho=0.99994
[getResults] pressure1=0.0747529; pressure2=-0.0214879; pressureDrop=0.0962408; drag=3.0532; lift=-0.39655
[Timer] step=31800; percent=99.375; passedTime=69.169; remTime=0.435025; MLUPs=13.7336
[LatticeStatistics] step=31800; t=15.9; uMax=0.0432243; avEnergy=0.000262978; avRho=0.999945
[getResults] pressure1=0.0749513; pressure2=-0.0215298; pressureDrop=0.0964811; drag=3.05908; lift=-0.453635
[Timer]
[Timer] ----------------Summary:Timer----------------
[Timer] measured time (rt) : 69.546s
[Timer] measured time (cpu): 68.938s
```

```
[CuboidGeometry2D]  Number of Cuboids:  7
[CuboidGeometry2D]  Delta (min):              0.005
[CuboidGeometry2D]        (max):              0.005
[CuboidGeometry2D]  Ratio (min):              0.75
[CuboidGeometry2D]        (max):              3
[CuboidGeometry2D]  Nodes (min):              5292
[CuboidGeometry2D]        (max):              5292
[CuboidGeometry2D] --------------------------------
```

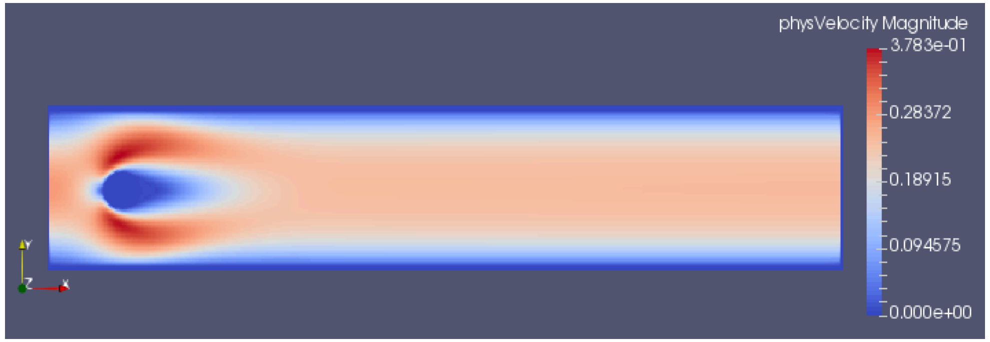
R_e=20 (laminar flow)

t = 0 s



t = 10 s

t = 5 s



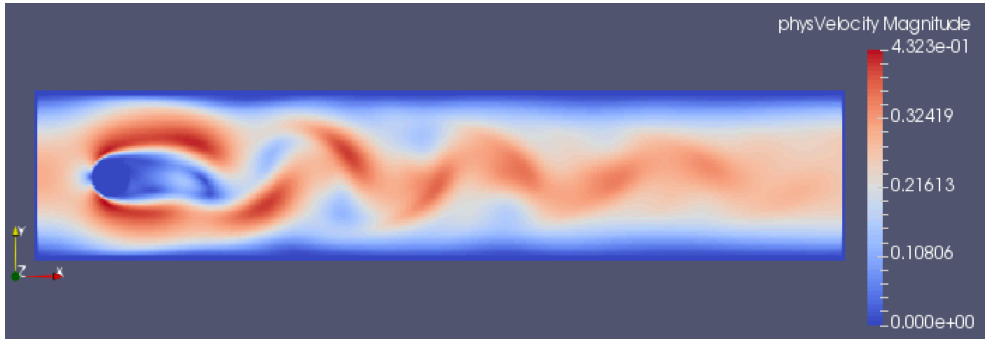
t = 15 s

R_e=100 (turbulence with Karman vortex street)

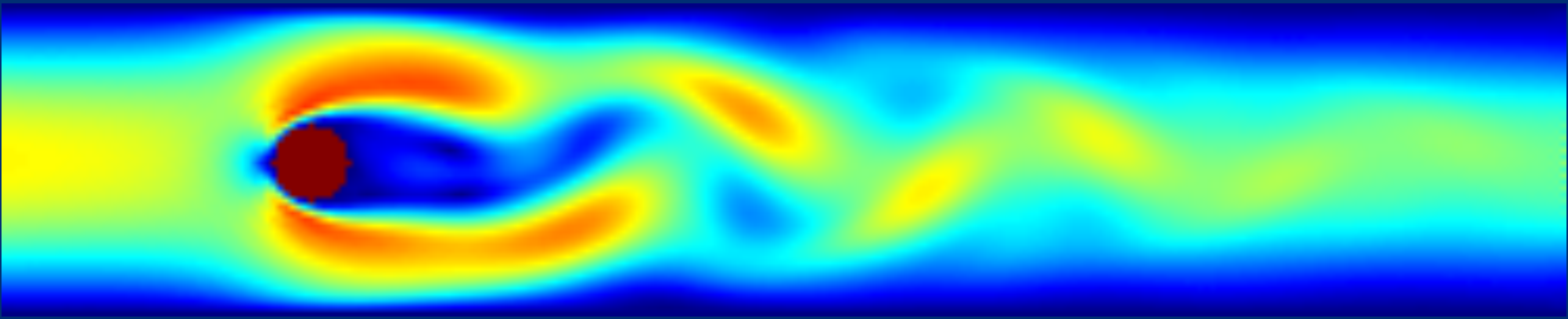
t = 0 s

t = 10 s

t = 5 s



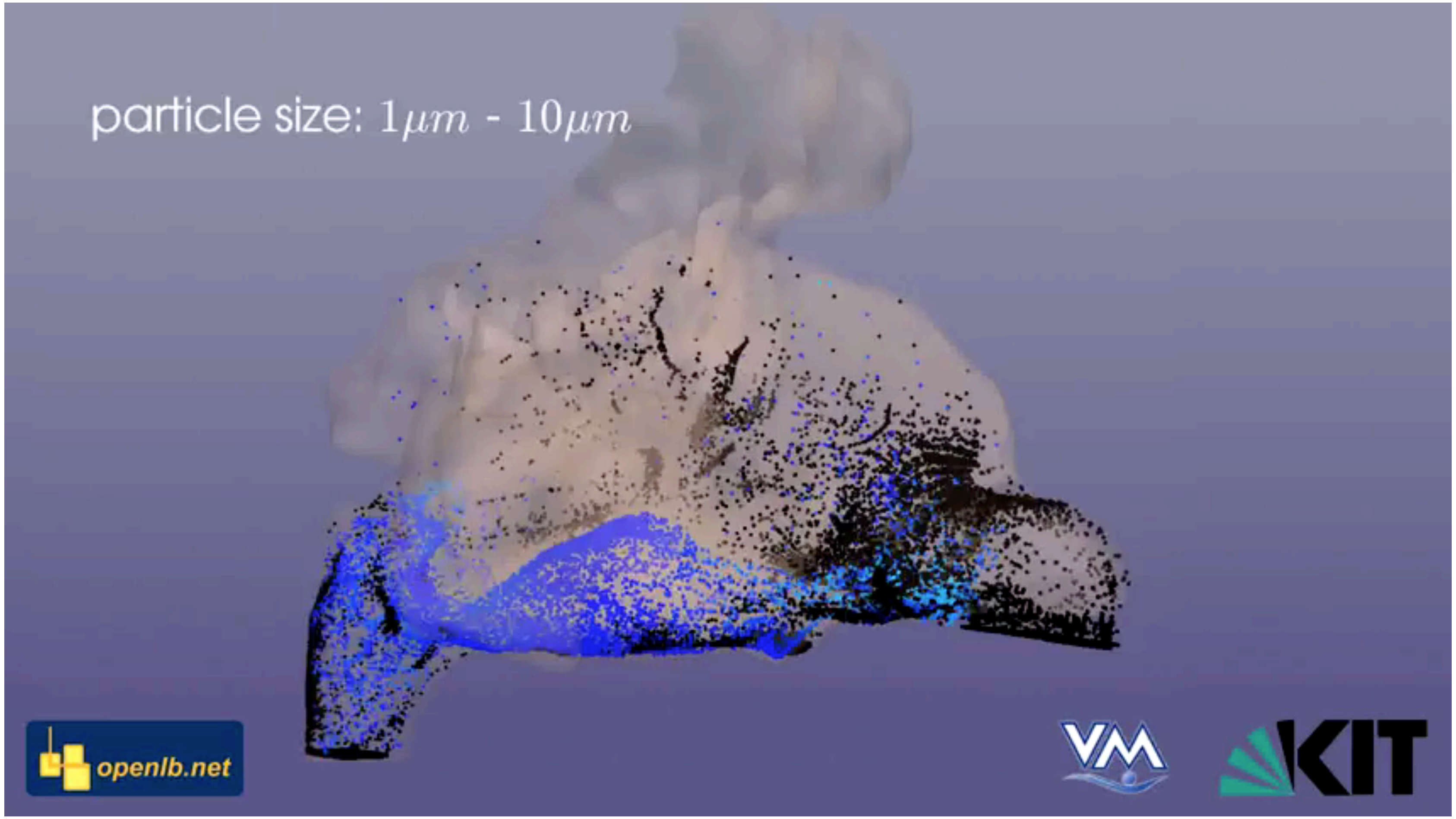
t = 15 s



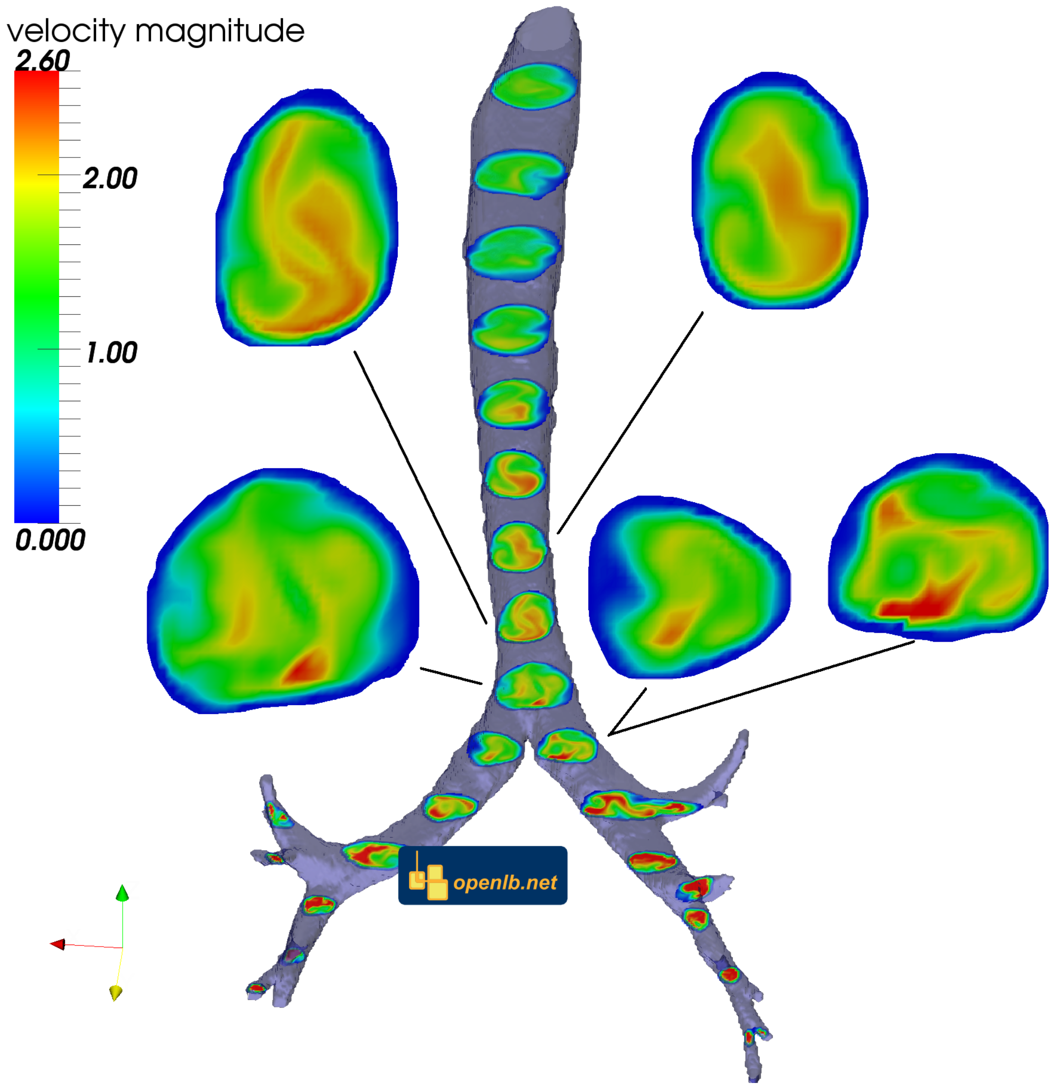
Example applications

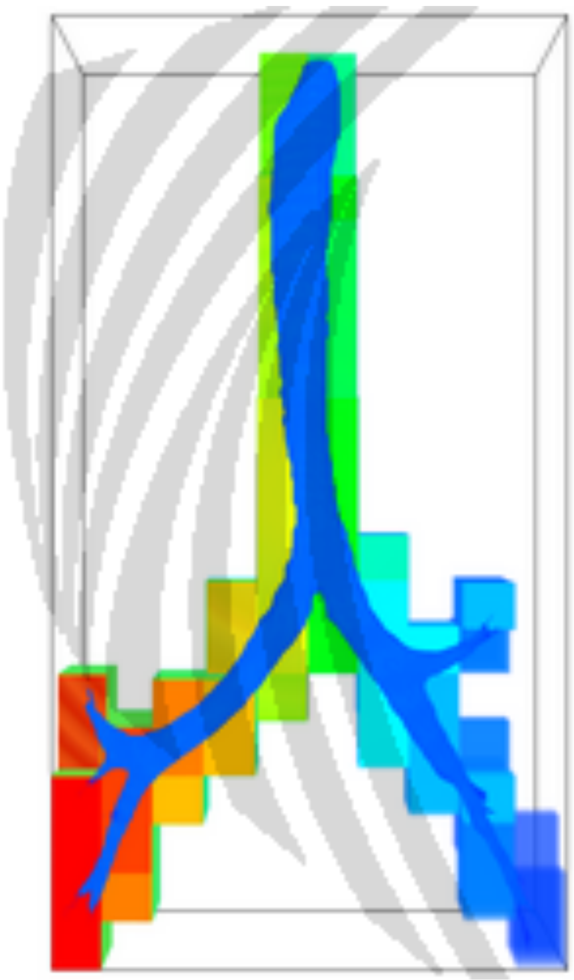
Rayleigh-Benard flow

Flow of particulates through nasal cavity

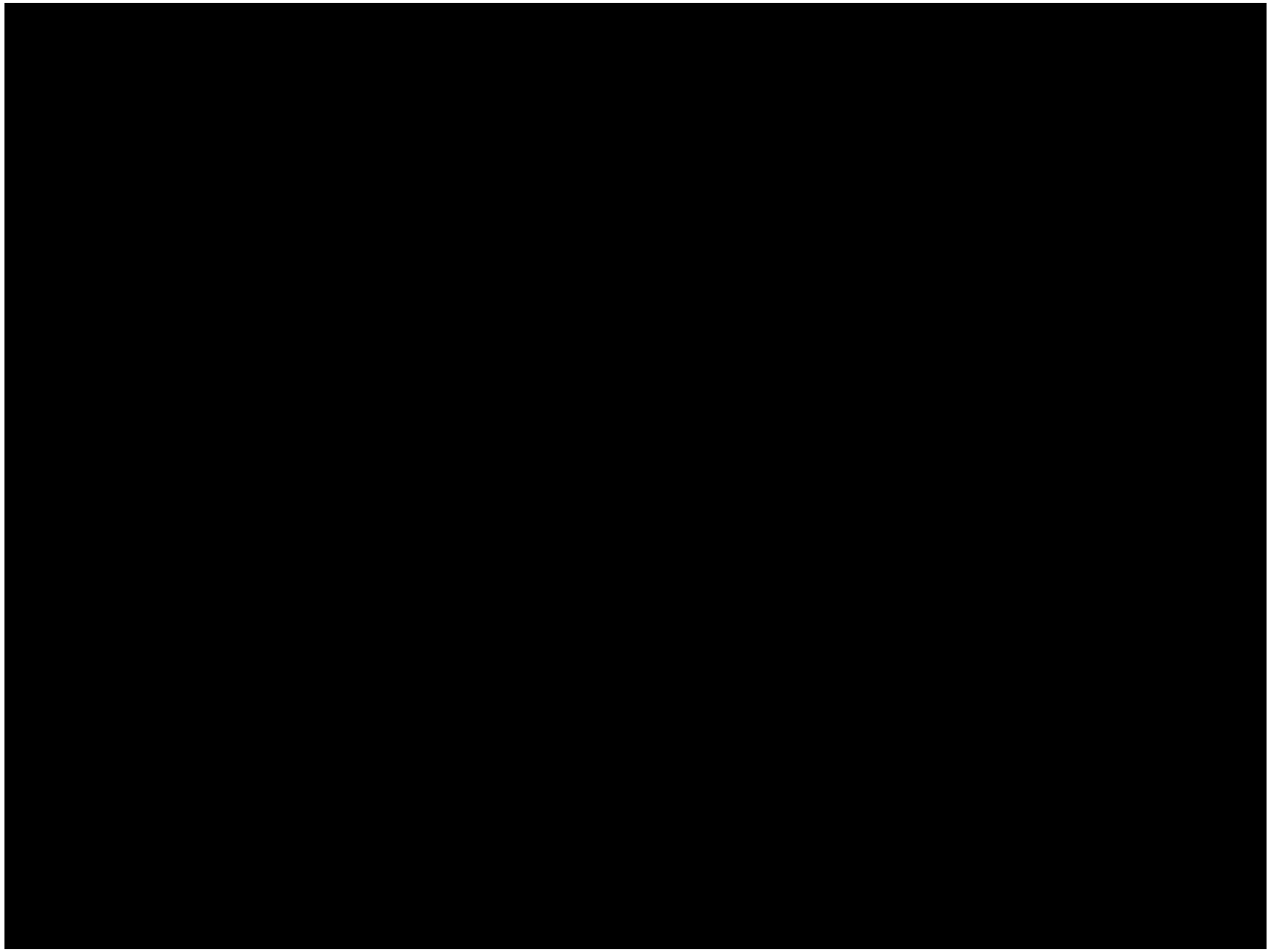


Flow through lungs - parallel processing





Turbulent flow in volcanoes



Our favorite – flow in porous media