Modelling multiphase flows/heat transfer/epidemics with Lattice Boltzmann Method

Grzegorz Gruszczyński[a,b], Michał Dzikowski[b], Łukasz Łaniewski-Wołłk[a,c]

25th November, 2021

[a] Politechnika Warszawska, Wydział Mechaniczny Energetyki i Lotnictwa

[b] Uniwersytet Warszawski, Interdyscyplinarne Centrum Modelowania Matematycznego i Komputerowego

[c] School of Mechanical and Mining Engineering, The University of Queensland, Australia

Table of contents

Introduction

LBM: Theory & Algorithm

- Discrete Boltzmann equation
- LBM - Algorithm
- Theory - deeper dive

LBM - Applications

- Heat Transfer in LBM
- Multiphase flows
- Epidemic modelling

(Un)structured Mesh

Memory requirements

Questions?

Introduction

Fluid dynamics and ...

What do Navier-Stokes, multiphase flows, enthalpy balance and epidemic modelling equations have in common?

Fluid dynamics and ...

What do Navier-Stokes, multiphase flows, enthalpy balance and epidemic modelling equations have in common?

Hydrodynamics: The continuity and momentum equations,

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \boldsymbol{u} = 0 \\ \rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\nabla p + \nabla \cdot \left(\mu [\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top] \right) + \boldsymbol{F} \end{cases}$$

Fluid dynamics and ...

What do Navier-Stokes, multiphase flows, enthalpy balance and epidemic modelling equations have in common?

Hydrodynamics: The continuity and momentum equations,

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \boldsymbol{u} = 0 \\ \rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\nabla p + \nabla \cdot \left(\mu [\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top] \right) + \boldsymbol{F} \end{cases}$$

Heat transfer: The Enthalpy balance equation,

$$\frac{\partial}{\partial t}(\rho c_p T) + \nabla \cdot (\boldsymbol{u} \rho c_p T) = \nabla \cdot (k \nabla T) + \dot{q}$$

Fluid dynamics and ...

What do Navier-Stokes, multiphase flows, enthalpy balance and epidemic modelling equations have in common?

Hydrodynamics: The continuity and momentum equations,

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \boldsymbol{u} = 0 \\ \rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\nabla p + \nabla \cdot \left(\mu [\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top] \right) + \boldsymbol{F} \end{cases}$$

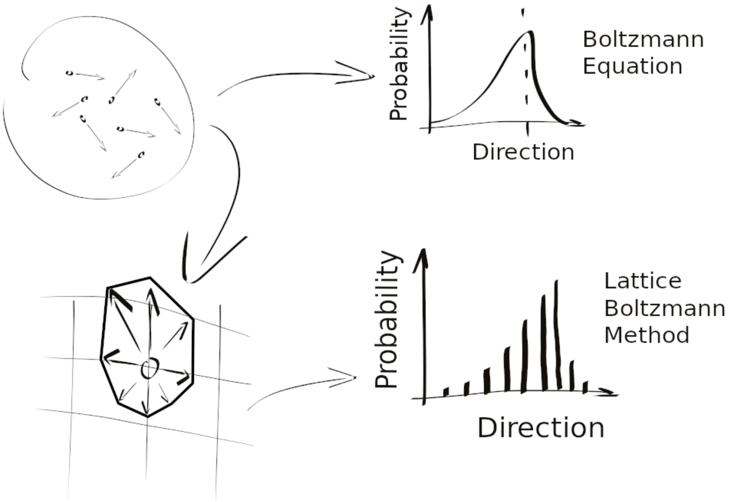
Heat transfer: The Enthalpy balance equation,

$$\frac{\partial}{\partial t}(\rho c_p T) + \nabla \cdot (\boldsymbol{u} \rho c_p T) = \nabla \cdot (k \nabla T) + \dot{q}$$

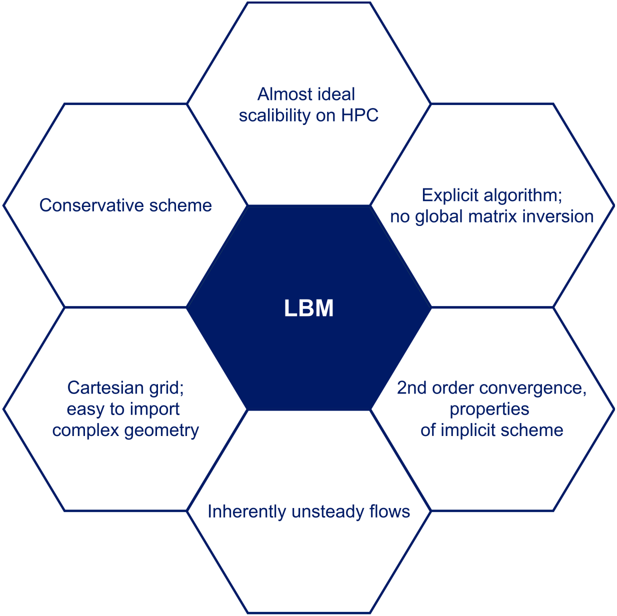
Multiphase flows: Phase field evolution equation,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \phi \mathbf{u} = \nabla \cdot M \left(\nabla \phi - \frac{\nabla \phi}{|\nabla \phi|} \frac{[1 - 4(\phi - \phi_0)^2]}{\gamma} \right)$$

What is Lattice Boltzmann Method?



Why Lattice Boltzmann Method?



LBM: Theory & Algorithm

LBM - Theory

Probability of finding a particle in the phase space:

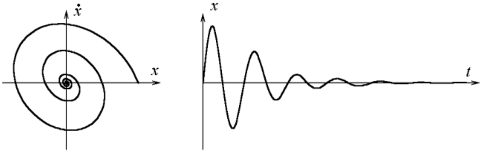


Figure 1: $\Psi = \Psi(t, x, \dot{x})$

Probability of finding a particle in the phase space:

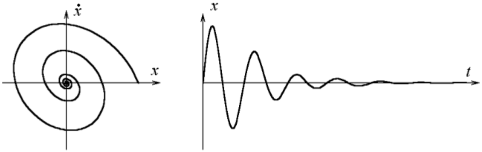


Figure 1: $\Psi = \Psi(t, x, \dot{x})$

In an infinitesimally small volume of the phase space $d\boldsymbol{x}d\boldsymbol{u}$:

$$\Psi_{no\ collisions}(t + dt, \boldsymbol{x} + d\boldsymbol{x}, \boldsymbol{u} + d\boldsymbol{u})d\boldsymbol{x}d\boldsymbol{u} = \Psi_{no\ collisions}(t, \boldsymbol{x}, \boldsymbol{u})d\boldsymbol{x}d\boldsymbol{u}$$

Probability of finding a particle in the phase space:

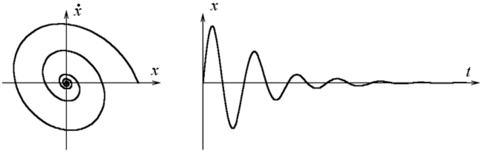


Figure 1: $\Psi = \Psi(t, \boldsymbol{x}, \dot{x})$

In an infinitesimally small volume of the phase space $d\boldsymbol{x}d\boldsymbol{u}$:

$$\Psi_{no\ collisions}(t + dt, \boldsymbol{x} + d\boldsymbol{x}, \boldsymbol{u} + d\boldsymbol{u})d\boldsymbol{x}d\boldsymbol{u} = \Psi_{no\ collisions}(t, \boldsymbol{x}, \boldsymbol{u})d\boldsymbol{x}d\boldsymbol{u}$$

Now, include the collision term $\mathbb{C}(\Psi)$:

$$\Psi(t + dt, \boldsymbol{x} + d\boldsymbol{x}, \boldsymbol{u} + d\boldsymbol{u})d\boldsymbol{x}d\boldsymbol{u} = \Psi(t, \boldsymbol{x}, \boldsymbol{u})d\boldsymbol{x}d\boldsymbol{u} + \mathbb{C}(\Psi)d\boldsymbol{x}d\boldsymbol{u}dt$$

Taylor series expansion:

$$\Psi(t+dt, \boldsymbol{x}+d\boldsymbol{x}, \boldsymbol{u}+d\boldsymbol{u}) = \Psi(t, \boldsymbol{x}, \boldsymbol{u}) + \frac{\partial \Psi}{\partial t} dt + \nabla_{\boldsymbol{x}} \Psi d\boldsymbol{x} + \nabla_{\boldsymbol{u}} \Psi d\boldsymbol{u}$$

Taylor series expansion:

$$\Psi(t+dt,\boldsymbol{x}+d\boldsymbol{x},\boldsymbol{u}+d\boldsymbol{u}) = \Psi(t,\boldsymbol{x},\boldsymbol{u}) + \frac{\partial \Psi}{\partial t}dt + \nabla_{\boldsymbol{x}}\Psi d\boldsymbol{x} + \nabla_{\boldsymbol{u}}\Psi d\boldsymbol{u}$$

Plug in:

$$\left[\Psi(t,\boldsymbol{x},\boldsymbol{u}) + \frac{\partial \Psi}{\partial t}dt + \nabla_{\boldsymbol{x}}\Psi d\boldsymbol{x} + \nabla_{\boldsymbol{u}}\Psi d\boldsymbol{u}\right] d\boldsymbol{x}d\boldsymbol{u} = \left[\Psi(t,\boldsymbol{x},\boldsymbol{u}) + \mathbb{C}(\Psi)dt\right] d\boldsymbol{x}d\boldsymbol{u}$$

Taylor series expansion:

$$\Psi(t+dt,\boldsymbol{x}+d\boldsymbol{x},\boldsymbol{u}+d\boldsymbol{u}) = \Psi(t,\boldsymbol{x},\boldsymbol{u}) + \frac{\partial\Psi}{\partial t}dt + \nabla_{\boldsymbol{x}}\Psi d\boldsymbol{x} + \nabla_{\boldsymbol{u}}\Psi d\boldsymbol{u}$$

Plug in:

$$\left[\Psi(t,\boldsymbol{x},\boldsymbol{u}) + \frac{\partial\Psi}{\partial t}dt + \nabla_{\boldsymbol{x}}\Psi d\boldsymbol{x} + \nabla_{\boldsymbol{u}}\Psi d\boldsymbol{u}\right] d\boldsymbol{x}d\boldsymbol{u} = \left[\Psi(t,\boldsymbol{x},\boldsymbol{u}) + \mathbb{C}(\Psi)dt\right] d\boldsymbol{x}d\boldsymbol{u}$$

Reformulate velocity $\boldsymbol{u} = \frac{d\boldsymbol{x}}{dt}$ and acceleration $\frac{d\boldsymbol{u}}{dt} = \frac{\boldsymbol{F}}{\rho}$:

$$\frac{\partial\Psi}{\partial t} + (\boldsymbol{u}\cdot\nabla_{\boldsymbol{x}})\Psi + (\frac{\boldsymbol{F}}{\rho}\cdot\nabla_{\boldsymbol{u}})\Psi = \mathbb{C}(\Psi)$$

Streaming and Collision:

$$\underbrace{\Psi(t+dt,\boldsymbol{x}+d\boldsymbol{x},\boldsymbol{u}+d\boldsymbol{u})d\boldsymbol{x}d\boldsymbol{u}}_{\textit{Streaming}} = \underbrace{\Psi(t,\boldsymbol{x},\boldsymbol{u})d\boldsymbol{x}d\boldsymbol{u} + \mathbb{C}(\Psi)d\boldsymbol{x}d\boldsymbol{u}dt}_{\textit{Collision}}$$

The Boltzmann equation can be viewed as a substantial derivative (of an intensive quantity Ψ) which is equal to the collision term $\mathbb{C}$ applied to the distribution function of Ψ:

$$\frac{\partial \Psi}{\partial t} + (\boldsymbol{u} \cdot \nabla_{\boldsymbol{x}})\Psi + (\frac{\boldsymbol{F}}{\rho} \cdot \nabla_{\boldsymbol{u}})\Psi = \mathbb{C}(\Psi)$$

LBM - Algorithm

$$\underbrace{f_i(\boldsymbol{x} + \boldsymbol{e_i}\Delta\boldsymbol{x}, t + \Delta t)}_{\textit{Streaming}} = \underbrace{f_i(\boldsymbol{x}, t) - \frac{1}{\tau}(f_i - f_i^{eq}) + F_i(\boldsymbol{x}, t)}_{\textit{Collision}}$$

- $\tau = \tau(\nu)$ relaxation parameter, ν is the kinematic viscosity
- f_i - discrete probability distribution function
- F_i - source term (ex. gravity force)

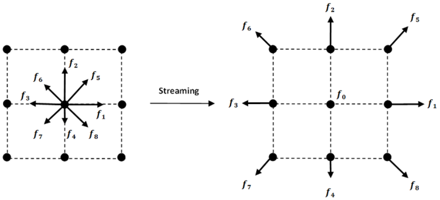


Figure 2: D2Q9: Streaming

Algorithm: Fluid

1 Initialize f_i^{in}

Algorithm: Fluid

1 Initialize f_i^{in}

2 Compute

$\rho = \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)$ *and* $\boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho} \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)\boldsymbol{e}_i + \frac{\boldsymbol{F}}{2\rho}\delta t$

Algorithm: Fluid

1 Initialize f_i^{in}

2 Compute
$\rho = \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)$ *and* $\boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho} \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)\boldsymbol{e}_i + \frac{\boldsymbol{F}}{2\rho}\delta t$

3 Compute $f_i^{eq}(\boldsymbol{x}, t) = w_i \rho \left[1 + \frac{\boldsymbol{e}_i \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_i \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}^2}{2c_s^2}\right]$ where $c_s^2 = \frac{1}{3}$

Algorithm: Fluid

1 Initialize f_i^{in}

2 Compute
$\rho = \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)$ *and* $\boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho} \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)\boldsymbol{e}_i + \frac{\boldsymbol{F}}{2\rho}\delta t$

3 Compute $f_i^{eq}(\boldsymbol{x}, t) = w_i \rho \left[1 + \frac{\boldsymbol{e}_i \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_i \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}^2}{2c_s^2}\right]$ where $c_s^2 = \frac{1}{3}$

4 Collision $f_i^{out}(\boldsymbol{x}, t) = f_i^{in}(\boldsymbol{x}, t) - \frac{1}{\tau_f}\left[f_i^{in}(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)\right] + F_i(\boldsymbol{x}, t)$

Algorithm: Fluid

1 Initialize f_i^{in}

2 Compute
$\rho = \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)$ *and* $\boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho} \sum_{i=0}^{8} f_i^{in}(\boldsymbol{x}, t)\boldsymbol{e}_i + \frac{\boldsymbol{F}}{2\rho}\delta t$

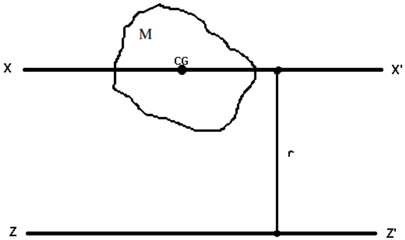
3 Compute $f_i^{eq}(\boldsymbol{x}, t) = w_i \rho \left[1 + \frac{\boldsymbol{e}_i \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_i \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}^2}{2c_s^2}\right]$ where $c_s^2 = \frac{1}{3}$

4 Collision $f_i^{out}(\boldsymbol{x}, t) = f_i^{in}(\boldsymbol{x}, t) - \frac{1}{\tau_f}\left[f_i^{in}(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)\right] + F_i(\boldsymbol{x}, t)$

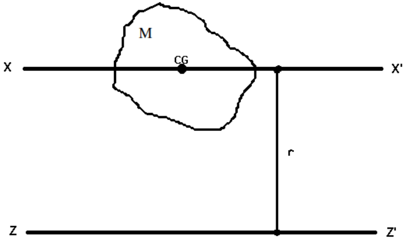
5 Streaming $f_i^{in}(\boldsymbol{x} + \boldsymbol{e}_i, t + 1) = f_i^{out}(\boldsymbol{x}, t)$

Theory - deeper dive
Concept of (Central) Moments

'Statistical' refreshment



'Statistical' refreshment



$$m_0 = M = \int r^0 \rho(r) d\Omega$$

$$m_1 = \mu = \frac{1}{M} \int r^1 \rho(r) d\Omega$$

$$m_2 = I_{zz'} = \int r^2 \rho(r) d\Omega$$

$$\sigma^2 = I_{xx'} \int (r - \mu)^2 \rho(r) d\Omega$$

Moments

The raw moments and central moments:

$$\kappa_{mn} = \sum_i (e_{i,x})^m (e_{i,y})^n f_i$$

$$\tilde{\kappa}_{mn} = \sum_i (e_{i,x} - u_x)^m (e_{i,y} - u_y)^n f_i$$

The raw moments and central moments:

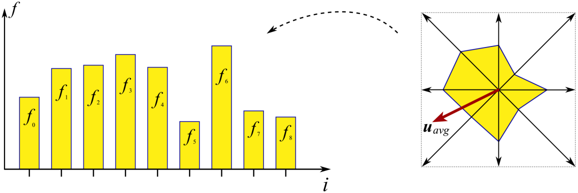
$$\kappa_{mn} = \sum_i (e_{i,x})^m (e_{i,y})^n f_i$$

$$\tilde{\kappa}_{mn} = \sum_i (e_{i,x} - u_x)^m (e_{i,y} - u_y)^n f_i$$

Physical interpretation:

$$\rho = \kappa_{00} = \sum_i f_i$$

$$\rho \boldsymbol{u} = \rho [u_x, u_y]^\top = [\kappa_{10}, \kappa_{01}]^\top = \sum_i f_i \boldsymbol{e}_i + \frac{\boldsymbol{F}}{2}\delta t$$



Alternatively, moments can be expressed in terms of matrix transformations:

$$\boldsymbol{\Upsilon} = \mathbb{M}\boldsymbol{f}$$
$$\tilde{\boldsymbol{\Upsilon}} = \mathbb{N}\boldsymbol{\Upsilon}$$

The resulting order of central moments is:

$$\tilde{\boldsymbol{\Upsilon}} = [\tilde{\Upsilon}_{00}, \tilde{\Upsilon}_{10}, \tilde{\Upsilon}_{01}, \tilde{\Upsilon}_{20}, \tilde{\Upsilon}_{02}, \tilde{\Upsilon}_{11}, \tilde{\Upsilon}_{21}, \tilde{\Upsilon}_{12}, \tilde{\Upsilon}_{22}]^{\top}$$

Equilibrium distribution function

The Maxwell-Boltzmann equilibrium distribution function in a continuous velocity space is known as:

$$\Psi^{M\text{-}B,\ eq} = \Psi^{M\text{-}B,\ eq}(\phi, \boldsymbol{\xi}, \boldsymbol{u}, \sigma) = \frac{\phi}{(2\pi\sigma^2)^{D/2}} exp\left[-\frac{(\boldsymbol{\xi} - \boldsymbol{u})^2}{2\sigma^2}\right]$$

where:

ϕ	– quantity of interest
$\boldsymbol{\xi}$	– microscopic 'particle' velocity
$\boldsymbol{u}$	– macroscopic 'flow' velocity
σ^2	– variance of the distribution

The continuous definition of the central moments is:

$$\tilde{\Upsilon}_{mn} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (\xi_x - u_x)^m (\xi_y - u_y)^n \Psi(\phi, \boldsymbol{\xi}, \boldsymbol{u}) d\xi_x d\xi_y$$

Algorithm fluid - revisited

1 Initialize f_i^{in}

2 Compute $\boldsymbol{u} = [u_x, u_y]^\top = [\kappa_{10}, \kappa_{01}]^\top = \dfrac{1}{\rho}\sum_i f_i \boldsymbol{e}_i + \frac{\boldsymbol{F}}{2\rho}\delta t$

3 Compute
$\tilde{\boldsymbol{\Upsilon}}(\boldsymbol{x}, t) = \mathbb{N}\mathbb{M}\boldsymbol{f}(\boldsymbol{x}, t)$,
$\tilde{\boldsymbol{\Upsilon}}^{eq}(\boldsymbol{x}, t) = [\rho, 0, 0, \sigma\rho, \sigma\rho, 0, 0, 0, \sigma^2\rho]^\top$
$\tilde{\boldsymbol{F}}(\boldsymbol{x}, t) = [0, F_x/\rho, F_y/\rho, 0, 0, 0, \sigma F_y/\rho, \sigma F_x/\rho, 0]^\top$

4 Collision $\tilde{\boldsymbol{\Upsilon}}(\boldsymbol{x}, t)^\star = \tilde{\boldsymbol{\Upsilon}} - \mathbb{S}(\tilde{\boldsymbol{\Upsilon}} - \tilde{\boldsymbol{\Upsilon}}^{eq}) + (\mathbb{1} - \mathbb{S}/2)\tilde{\boldsymbol{F}}$

5 Streaming $f_i(\boldsymbol{x} + \boldsymbol{e}\delta t, t + \delta t) = \mathbb{M}^{-1}\mathbb{N}^{-1}\tilde{\boldsymbol{\Upsilon}}_i(\boldsymbol{x}, t)^\star$

LBM - Applications

Heat Transfer in LBM

Algorithm: Energy Field

'Advection - Diffusion' of H is solved on a separate D2Q9 lattice

1 Initialize $h_i^{in}(\boldsymbol{x}, t)$

2 Compute $H = \rho c_p T = \sum_{i=0}^{9} h_i^{in}(\boldsymbol{x}, t)$

3 Compute $h_i^{eq}(\boldsymbol{x}, t) = H w_i \left[1 + \frac{\boldsymbol{e}_i \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_i \boldsymbol{u})^2}{2c_s^2} - \frac{\boldsymbol{u}^2}{2c_s^2} \right]$

4 Collision $h_i^{out}(\boldsymbol{x}, t) = h_i^{in}(\boldsymbol{x}, t) - \frac{1}{\tau_T} \left[h_i^{in}(\boldsymbol{x}, t) - h_i^{eq}(\boldsymbol{x}, t) \right] + \frac{\dot{q}}{\rho c_p}$

5 Streaming $h_i^{in}(\boldsymbol{x} + \boldsymbol{e}_i, t + 1) = h_i^{out}(\boldsymbol{x}, t)$

Now, the temperature field can be solved in a fluid:

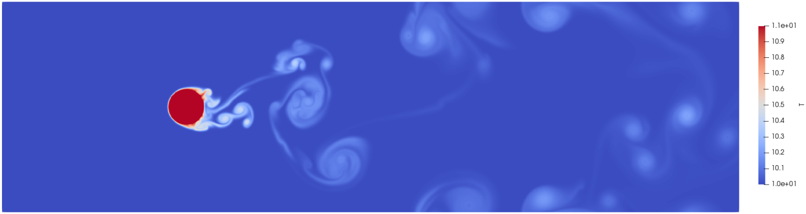


Figure 3: Re $=$ 1000, Pr $=$ 0.71, D $=$ 128 [lu]

Multiphase flows

The separation flux, $\mathbf{j}_S$, is supposed to counteract the diffusion and reach a predefined interface profile in the equilibrium state:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \phi \mathbf{u} = \nabla \cdot (\underbrace{M \nabla \phi}_{\mathbf{j}_D} - \mathbf{j}_S).$$

Advection-Diffusion Equation revisited

The separation flux, $\mathbf{j}_S$, is supposed to counteract the diffusion and reach a predefined interface profile in the equilibrium state:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \phi \mathbf{u} = \nabla \cdot (\underbrace{M \nabla \phi}_{\mathbf{j}_D} - \mathbf{j}_S).$$

To obtain the anti-diffusive effect, we require:

$$\mathbf{j}_S = \mathbf{j}_D^{eq} = M \nabla \phi^{eq}.$$

The separation flux, $\mathbf{j}_S$, is supposed to counteract the diffusion and reach a predefined interface profile in the equilibrium state:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \phi \mathbf{u} = \nabla \cdot (\underbrace{M \nabla \phi}_{\mathbf{j}_D} - \mathbf{j}_S).$$

To obtain the anti-diffusive effect, we require:

$$\mathbf{j}_S = \mathbf{j}_D^{eq} = M \nabla \phi^{eq}.$$

Let us use a *tanh* to smooth the step interface,

$$\phi^{eq} = \frac{1}{2} tanh \left(\frac{2(\mathbf{x} - \mathbf{x}_0)}{\gamma} \right),$$

where γ is the thickness of the interface.

Evaluate diffusive flux in the equilibrium:

$$\mathbf{j}_D^{eq} = M\nabla \overbrace{\left[\frac{1}{2} tanh\left(\frac{2(\mathbf{x}-\mathbf{x}_0)}{\gamma}\right)\right]}^{\phi^{eq}} = \frac{M}{2}\mathbf{n}\frac{\partial}{\partial \mathbf{x}_n} tanh\left[\left(\frac{2(\mathbf{x}-\mathbf{x}_0)}{\gamma}\right)\right]$$

Evaluate diffusive flux in the equilibrium:

$$\mathbf{j}_D^{eq} = M\nabla \overbrace{\left[\frac{1}{2} tanh\left(\frac{2(\mathbf{x}-\mathbf{x}_0)}{\gamma}\right)\right]}^{\phi^{eq}} = \frac{M}{2}\mathbf{n}\frac{\partial}{\partial \mathbf{x}_n} tanh\left[\left(\frac{2(\mathbf{x}-\mathbf{x}_0)}{\gamma}\right)\right]$$

$$= \frac{M}{\gamma}\mathbf{n}\underbrace{\left[1 - tanh^2\left(\frac{2(\mathbf{x}-\mathbf{x}_0)}{\gamma}\right)\right]}_{1-4(\phi^{eq})^2} = M\mathbf{n}\frac{1-4(\phi^{eq})^2}{\gamma}$$

Evaluate diffusive flux in the equilibrium:

$$\mathbf{j}_D^{eq} = M\nabla \overbrace{\left[\frac{1}{2} tanh\left(\frac{2(\mathbf{x}-\mathbf{x}_0)}{\gamma}\right)\right]}^{\phi^{eq}} = \frac{M}{2}\mathbf{n}\frac{\partial}{\partial \mathbf{x}_n} tanh\left[\left(\frac{2(\mathbf{x}-\mathbf{x}_0)}{\gamma}\right)\right]$$

$$= \frac{M}{\gamma}\mathbf{n}\underbrace{\left[1 - tanh^2\left(\frac{2(\mathbf{x}-\mathbf{x}_0)}{\gamma}\right)\right]}_{1-4(\phi^{eq})^2} = M\mathbf{n}\frac{1-4(\phi^{eq})^2}{\gamma}$$

Therefore:

$$\mathbf{j}_S = M\mathbf{n}\frac{1-4\phi^2}{\gamma} \qquad where \quad \mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$$

Phase field

Phase field evolution equation:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \phi \mathbf{u} = \nabla \cdot M \left(\nabla \phi - \frac{\nabla \phi}{|\nabla \phi|} \frac{[1 - 4(\phi - \phi_0)^2]}{\gamma} \right)$$

Interface location: $\phi_0 = (\phi_H + \phi_L)/2$.

Density can be calculated using a linear interpolation between ρ_H and ρ_L,

$$\rho = \rho_L + \frac{\phi - \phi_L}{\phi_H - \phi_L} (\rho_H - \rho_L).$$

Epidemic modelling

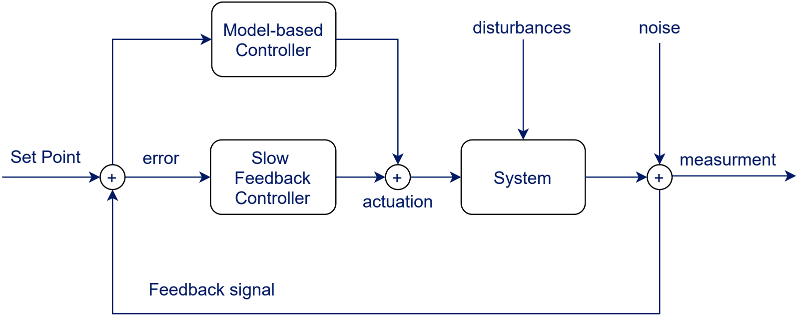


Figure 4: Control Theory and Covid-19 [1].

There is a trade-off between amount of control variables and accuracy.

- 0D system of SIR PDE (Suspectible, Infected, Recovered)
- Cellular automata
- **Spatial SIR**
- Agent Models
- Machine Learning

The greatest challenge is the input data and calibration of the model.

Simulate a SIR-like, spatial epidemic model with varying population density.

Reminding the Enthalpy balance equation,

$$\frac{\partial}{\partial t}(\rho c_p T) + \nabla \cdot (\boldsymbol{u}\rho c_p T) = \nabla \cdot (k \nabla T) + \dot{q} \tag{1}$$

The set of SIR equations can be viewed as a variation of eq. (1):

$$\frac{\partial \rho T}{\partial t} = \nabla \cdot k_T \nabla T + \dot{q}_T \tag{2}$$

where T is the fraction of population, $T \in \{s, i, r\}$, k_T is the diffusivity coefficient and $\dot{q}_T$ is a source term which couples the equations.

(Naive[1]) Spatial SIR model

$$\frac{\partial \rho T}{\partial t} = \nabla \cdot k_T \nabla T + \dot{q}_T$$

can be expanded as,

$$\begin{cases} \frac{\partial}{\partial t}\rho(\boldsymbol{x})s(t,\boldsymbol{x}) & = \nabla \cdot k_s \nabla s(t,\boldsymbol{x}) - \beta s(t,\boldsymbol{x}) i(t,\boldsymbol{x}) \\ \frac{\partial}{\partial t}\rho(\boldsymbol{x})i(t,\boldsymbol{x}) & = \nabla \cdot k_i \nabla i(t,\boldsymbol{x}) + \beta s(t,\boldsymbol{x}) i(t,\boldsymbol{x}) - \gamma i(t,\boldsymbol{x}) \\ \frac{\partial}{\partial t}\rho(\boldsymbol{x})r(t,\boldsymbol{x}) & = \nabla \cdot k_r \nabla r(t,\boldsymbol{x}) + \gamma i(t,\boldsymbol{x}) \end{cases}$$

where:

β - frequency of contacts,

γ - frequency of recoveries.

1

(Naive[1]) Spatial SIR model

$$\frac{\partial \rho T}{\partial t} = \nabla \cdot k_T \nabla T + \dot{q}_T$$

can be expanded as,

$$\begin{cases} \frac{\partial}{\partial t}\rho(\boldsymbol{x})s(t,\boldsymbol{x}) & = \nabla \cdot k_s \nabla s(t,\boldsymbol{x}) - \beta s(t,\boldsymbol{x}) i(t,\boldsymbol{x}) \\ \frac{\partial}{\partial t}\rho(\boldsymbol{x})i(t,\boldsymbol{x}) & = \nabla \cdot k_i \nabla i(t,\boldsymbol{x}) + \beta s(t,\boldsymbol{x}) i(t,\boldsymbol{x}) - \gamma i(t,\boldsymbol{x}) \\ \frac{\partial}{\partial t}\rho(\boldsymbol{x})r(t,\boldsymbol{x}) & = \nabla \cdot k_r \nabla r(t,\boldsymbol{x}) + \gamma i(t,\boldsymbol{x}) \end{cases}$$

where:

β - frequency of contacts,

γ - frequency of recoveries.

1

Modelling equations

(Naive[1]) Spatial SIR model

$$\frac{\partial \rho T}{\partial t} = \nabla \cdot k_T \nabla T + \dot{q}_T$$

can be expanded as,

$$\begin{cases} \frac{\partial}{\partial t}\rho(\boldsymbol{x})s(t,\boldsymbol{x}) & = \nabla \cdot k_s \nabla s(t,\boldsymbol{x}) - \beta s(t,\boldsymbol{x}) i(t,\boldsymbol{x}) \\ \frac{\partial}{\partial t}\rho(\boldsymbol{x})i(t,\boldsymbol{x}) & = \nabla \cdot k_i \nabla i(t,\boldsymbol{x}) + \beta s(t,\boldsymbol{x}) i(t,\boldsymbol{x}) - \gamma i(t,\boldsymbol{x}) \\ \frac{\partial}{\partial t}\rho(\boldsymbol{x})r(t,\boldsymbol{x}) & = \nabla \cdot k_r \nabla r(t,\boldsymbol{x}) + \gamma i(t,\boldsymbol{x}) \end{cases}$$

where:

- β - frequency of contacts,
- γ - frequency of recoveries.

1

(Naive[1]) Spatial SIR model

$$\frac{\partial \rho T}{\partial t} = \nabla \cdot k_T \nabla T + \dot{q}_T$$

can be expanded as,

$$\begin{cases} \frac{\partial}{\partial t}\rho(\boldsymbol{x})s(t,\boldsymbol{x}) & = \nabla \cdot k_s \nabla s(t,\boldsymbol{x}) - \beta s(t,\boldsymbol{x})i(t,\boldsymbol{x}) \\ \frac{\partial}{\partial t}\rho(\boldsymbol{x})i(t,\boldsymbol{x}) & = \nabla \cdot k_i \nabla i(t,\boldsymbol{x}) + \beta s(t,\boldsymbol{x})i(t,\boldsymbol{x}) - \gamma i(t,\boldsymbol{x}) \\ \frac{\partial}{\partial t}\rho(\boldsymbol{x})r(t,\boldsymbol{x}) & = \nabla \cdot k_r \nabla r(t,\boldsymbol{x}) + \gamma i(t,\boldsymbol{x}) \end{cases}$$

where:

β - frequency of contacts,

γ - frequency of recoveries.

[1] Cannot explain the spatial transmission by infection if individuals are at rest. Moreover, humans would move away from an increasing gradient of the s,i,r.

The disease may spread to neighbours with some probability $P(r)$, where r is the "infectious" distance. Let us define the **viral load** as $W = I \star P(r)$.

$$\frac{\partial}{\partial t} S = -\beta \frac{S}{N} W$$
$$\frac{\partial}{\partial t} I = \beta \frac{S}{N} W - \gamma I$$
$$\frac{\partial}{\partial t} R = \gamma I$$

The disease may spread to neighbours with some probability $P(r)$, where r is the "infectious" distance. Let us define the **viral load** as $W = I \star P(r)$.

$$\frac{\partial}{\partial t} S = -\beta \frac{S}{N} W$$
$$\frac{\partial}{\partial t} I = \beta \frac{S}{N} W - \gamma I$$
$$\frac{\partial}{\partial t} R = \gamma I$$

Substituting [4] $W \approx I + \frac{r^2}{8} \Delta I$,

$$\frac{\partial}{\partial t} S = -\beta \frac{S}{N} (I + \frac{r^2}{8} \Delta I)$$
$$\frac{\partial}{\partial t} I = \beta \frac{S}{N} (I + \frac{r^2}{8} \Delta I) - \gamma I$$
$$\frac{\partial}{\partial t} R = \gamma I$$

Notice that the diffusivity depends on S, which is decreasing in time.

Sample Implementation

Contributions to master, excluding merge commits and bot accounts

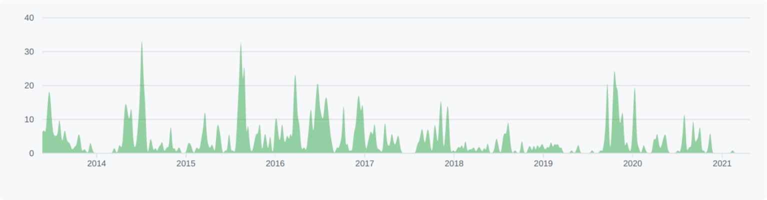


Figure 5: The TCLB solver [2, 3] is used (https://github.com/CFD-GO/TCLB/) . Is developed at WUT, ICM UW and UQ.

Sample simulations (with the naive spatial SIR model):

- Case_I - high diffusivity, slow incubation

  https://drive.google.com/open?id=12qOeuCx4YshsqIw16t4pjPjtlwYGqyDD

- Case_II - slow diffusivity, quick incubation

  https://drive.google.com/open?id=1xsiLQc_gsasAfcA6QcXBPrbXwVnnbD3C

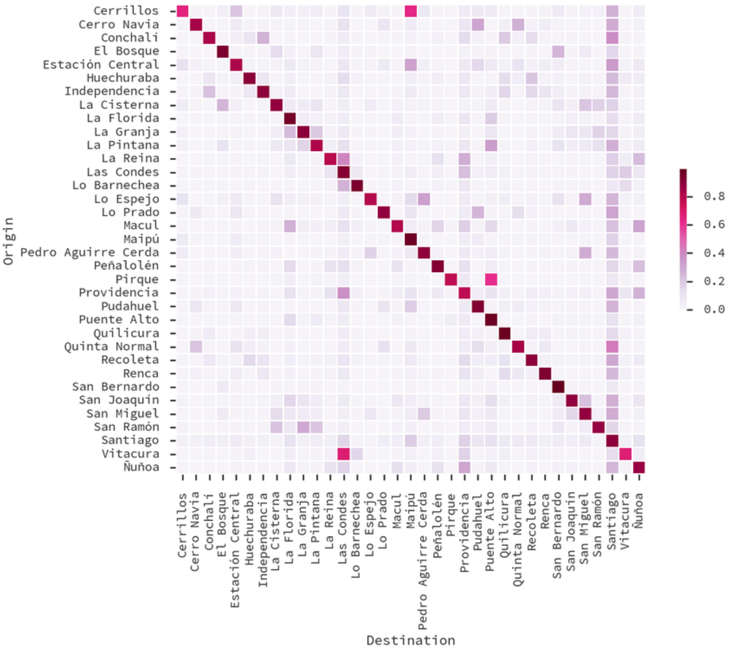
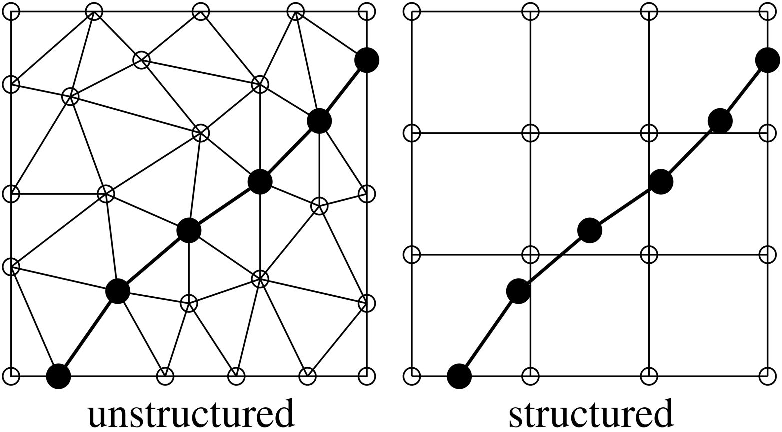
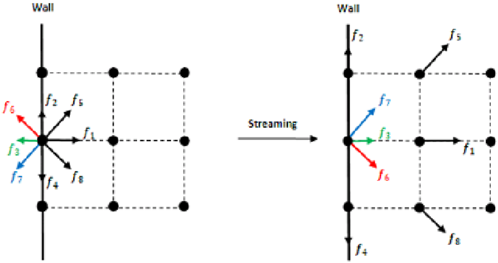


Figure 6: Example of an Origin-Destination matrix.

(Un)structured Mesh

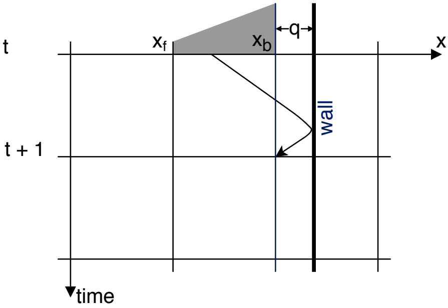
(Un)structured Mesh?





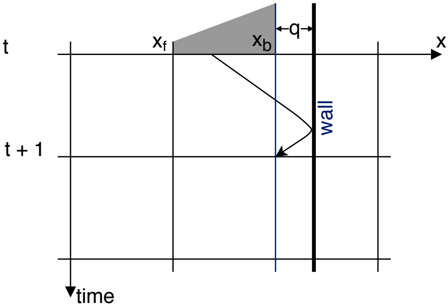
$$f_{\bar{i}}(\boldsymbol{x}_b, t + \Delta t) = f_i(\boldsymbol{x}_b, t)$$

It is assumed that during each streaming step, the population travels a distance $|\boldsymbol{e}_i|\Delta t$.



Interpolated Bounce Back Boundary Condition

It is assumed that during each streaming step, the population travels a distance $|\boldsymbol{e}_i|\Delta t$.



$$
f_{\bar{i}}(\boldsymbol{x}_b, t+\Delta t) = \begin{cases} 2q f_i^{\star}(\boldsymbol{x}_b, t) + (1-2q) f_i^{\star}(\boldsymbol{x}_f, t) & \text{for } q \in [0, 0.5]\,, \\ \frac{1}{2q} f_i^{\star}(\boldsymbol{x}_b, t) + \dfrac{2q-1}{2q} f_{\bar{i}}^{\star}(\boldsymbol{x}_b, t) & \text{for } q \in (0.5, 1]\,. \end{cases}
$$

Memory requirements

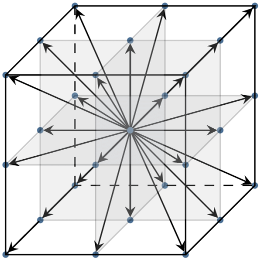


Figure 7: D3Q27-stencil

Memory requirements

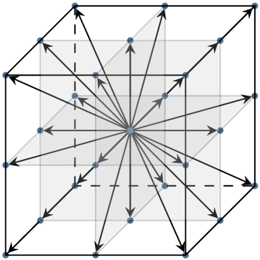


Figure 7: D3Q27-stencil

	No [-]	[double]	[Bytes]
DF	2x27	54	432
DF temp	2x27	54	432
q	2x27	13.5	108
flag	1	0.5	4
memory per node		122	976

Table 1: Theoretical memory requirements for: 3D, DDF model with interpolated BC.

Memory requirements

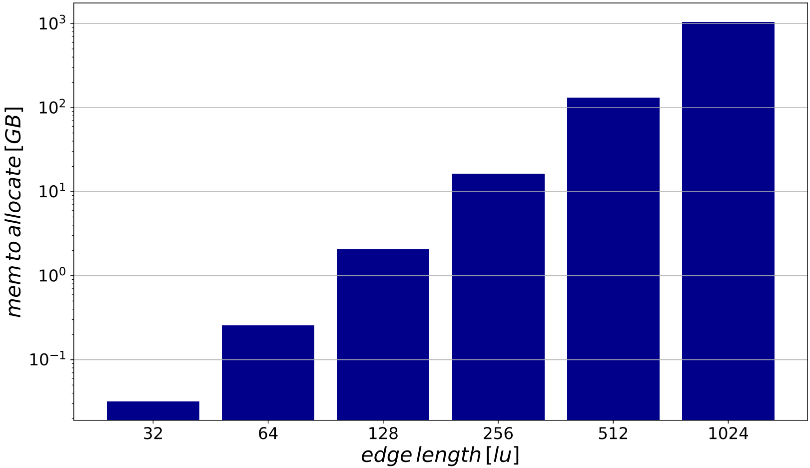


Figure 8: Theoretical memory requirements for a 3D cube domain, D3Q27Q27 lattice with interpolated BC.

Memory requirements

Q: Does the high memory requirement limits applicability of LBM?

Memory requirements

Q: Does the high memory requirement limits applicability of LBM?

Not necassary. Similar problems occur in other CFD methods like VOF, FEM, FD.

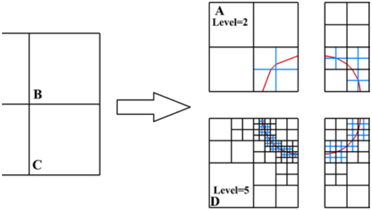


Figure 9: Mesh refinement.

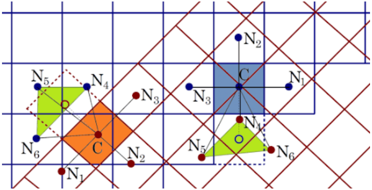


Figure 10: Overlapping mesh.

Questions?

References

- Steve Brunton. *Control Theory and COVID-19*. 2020. URL: https://youtu.be/BTLZu-1IMcE?list=PLMrJAkhIeNNR_s-TveQKkKpreIoe5L_uL (visited on 11/14/2021).
- Ł. Łaniewski-Wołłk and J. Rokicki. "Adjoint Lattice Boltzmann for topology optimization on multi-GPU architecture". *Computers and Mathematics with Applications* 71 (2016), pp. 833–848. DOI: 10.1016/j.camwa.2015.12.043.
- Łukasz Łaniewski-Wołłk et al. *CFD-GO/TCLB: Version 6.5*. Version v6.5.0. 2020. DOI: 10.5281/zenodo.4074541. URL: https://github.com/CFD-GO/TCLB.

Seong Hun Paeng and Jonggul Lee. "Continuous and discrete SIR-models with spatial distributions". *Journal of Mathematical Biology* 74.7 (2017), pp. 1709–1727. ISSN: 14321416. DOI: 10.1007/s00285-016-1071-8.