From Cellular Automata to the Lattice Boltzmann Method

Theresa Pollinger, Marcial Gaißert
Ferienakademie Sarntal 2017
September 9, 2017

Outline

Cellular Automata

Lattice Gas Automata

The Lattice Boltzmann Method

Cellular Automata

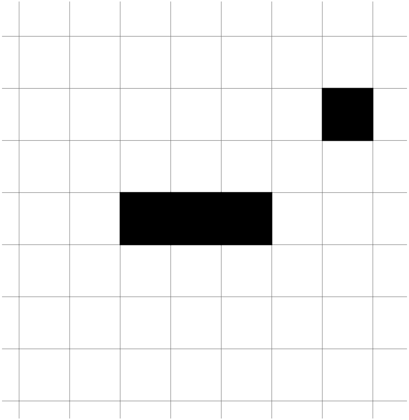
Cellular Automata

- regular arrangement of cells
- each with discrete state
- updated simultaneously at each step
- update rules:
  - same for each cell
  - only dependent on close-by cells

a grid of cells

Cellular Automata

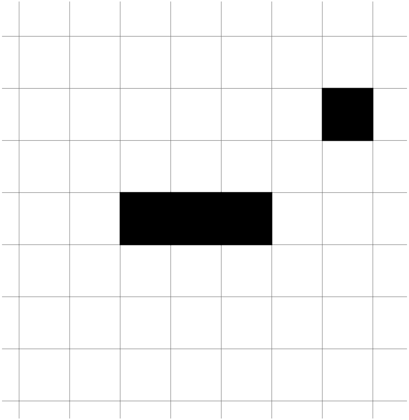
- regular arrangement of cells
- each with discrete state
- updated simultaneously at each step
- update rules:
  - same for each cell
  - only dependent on close-by cells



a grid of cells with state

Cellular Automata

- regular arrangement of cells
- each with discrete state
- updated simultaneously at each step
- update rules:
  - same for each cell
  - only dependent on close-by cells



a grid of cells with state

Cellular Automata

- regular arrangement of cells
- each with discrete state
- updated simultaneously at each step
- update rules:
  - same for each cell
  - only dependent on close-by cells

previous	#live neighbors	next
■	< 2	□
	$2, 3$	■
	> 3	□
□	3	■

Rules of Conway's Game of Life

Cellular Automata

- regular arrangement of cells
- each with discrete state
- updated simultaneously at each step
- update rules:
  - same for each cell
  - only dependent on close-by cells

previous	#live neighbors	next
■	< 2	□
	$2, 3$	■
	> 3	□
□	3	■

Rules of Conway's Game of Life

Cellular Automata

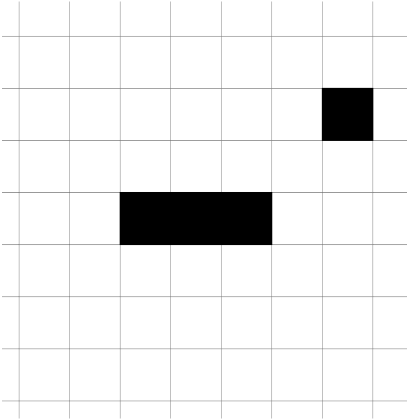
- regular arrangement of cells
- each with discrete state
- updated simultaneously at each step
- update rules:
  - same for each cell
  - only dependent on close-by cells

previous	#live neighbors	next
■	< 2	□
	$2, 3$	■
	> 3	□
□	3	■

Rules of Conway's Game of Life

Cellular Automata

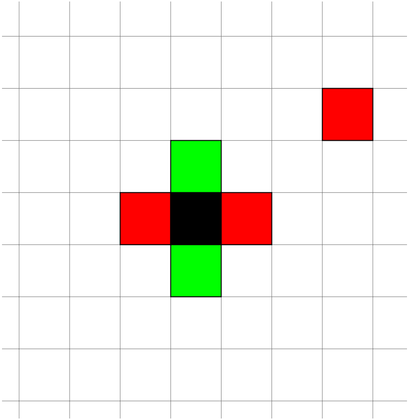
- regular arrangement of cells
- each with discrete state
- updated simultaneously at each step
- update rules:
  - same for each cell
  - only dependent on close-by cells



a grid of cells with state

Cellular Automata

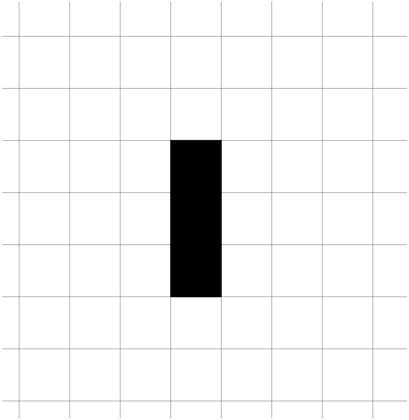
- regular arrangement of cells
- each with discrete state
- updated simultaneously at each step
- update rules:
  - same for each cell
  - only dependent on close-by cells



changes in first step

Cellular Automata

- regular arrangement of cells
- each with discrete state
- updated simultaneously at each step
- update rules:
  - same for each cell
  - only dependent on close-by cells



after first step

Cellular Automata – advantages and disadvantages

advantages	disadvantages
• simple structure • easy to implement • easy to parallelize	• difficult to design for given macroscopic behavior

Cellular Automata – advantages and disadvantages

advantages	disadvantages
• simple structure • easy to implement • easy to parallelize	• difficult to design for given macroscopic behavior

Cellular Automata – advantages and disadvantages

advantages	disadvantages
• simple structure • easy to implement • easy to parallelize	• difficult to design for given macroscopic behavior

Cellular Automata – advantages and disadvantages

advantages	disadvantages
• simple structure • easy to implement • easy to parallelize	• difficult to design for given macroscopic behavior

Cellular Automata – advantages and disadvantages

advantages	disadvantages
• simple structure • easy to implement • easy to parallelize	• difficult to design for given macroscopic behavior

Lattice Gas Automata

Lattice Gas Automata

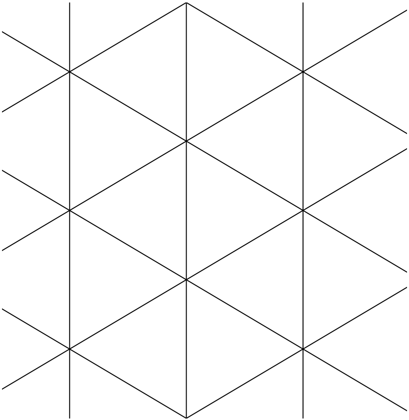
- Regular arrangement of nodes
  - HPP: square lattice $\rightarrow$ does not lead to Navier-Stokes
  - FHP: triangular lattice
- 6 (7) cells per node
  - one for each direction
- for each step, do:
  - collision
  - propagation

Lattice Gas Automata

- Regular arrangement of nodes
  - HPP: square lattice $\rightarrow$ does not lead to Navier-Stokes
  - FHP: triangular lattice
- 6 (7) cells per node
  - one for each direction
- for each step, do:
  - collision
  - propagation

Lattice Gas Automata

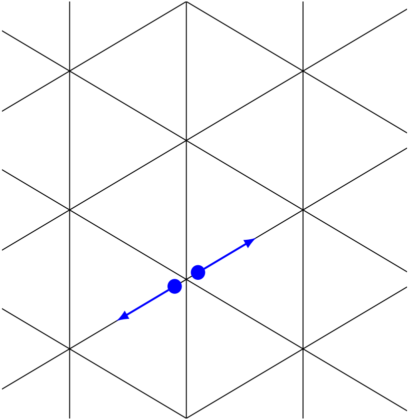
- Regular arrangement of nodes
  - HPP: square lattice $\rightarrow$ does not lead to Navier-Stokes
  - FHP: triangular lattice
- 6 (7) cells per node
  - one for each direction
- for each step, do:
  - collision
  - propagation



Triangular lattice

Lattice Gas Automata

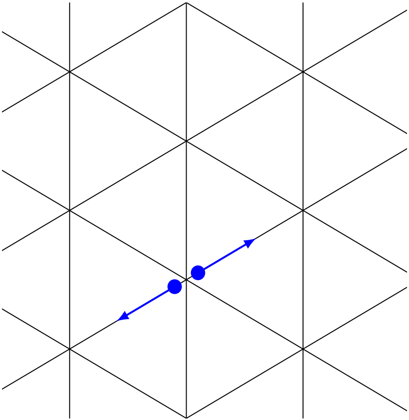
- Regular arrangement of nodes
  - HPP: square lattice $\rightarrow$ does not lead to Navier-Stokes
  - FHP: triangular lattice
- 6 (7) cells per node
  - one for each direction
- for each step, do:
  - collision
  - propagation



before collision

Lattice Gas Automata

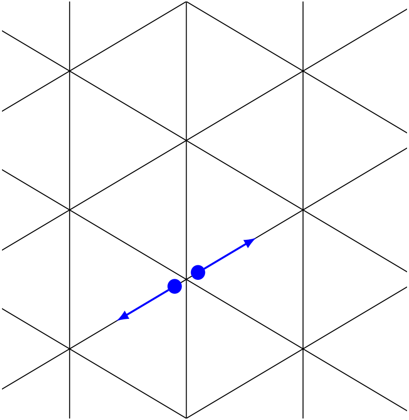
- Regular arrangement of nodes
  - HPP: square lattice
    $\rightarrow$ does not lead to Navier-Stokes
  - FHP: triangular lattice
- 6 (7) cells per node
  - one for each direction
- for each step, do:
  - collision
  - propagation



before collision

Lattice Gas Automata

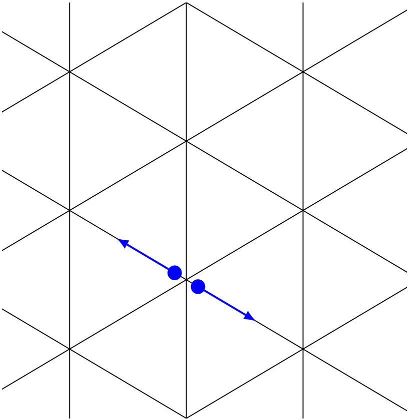
- Regular arrangement of nodes
  - HPP: square lattice $\rightarrow$ does not lead to Navier-Stokes
  - FHP: triangular lattice
- 6 (7) cells per node
  - one for each direction
- for each step, do:
  - collision
  - propagation



before collision

Lattice Gas Automata

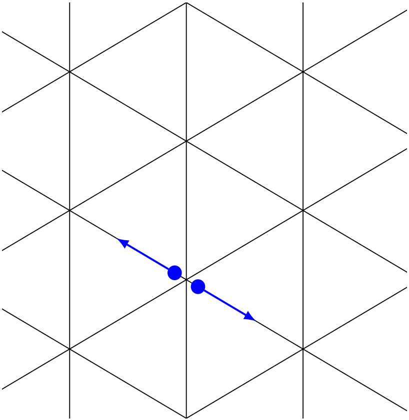
- Regular arrangement of nodes
  - HPP: square lattice $\rightarrow$ does not lead to Navier-Stokes
  - FHP: triangular lattice
- 6 (7) cells per node
  - one for each direction
- for each step, do:
  - collision
  - propagation



after collision

Lattice Gas Automata

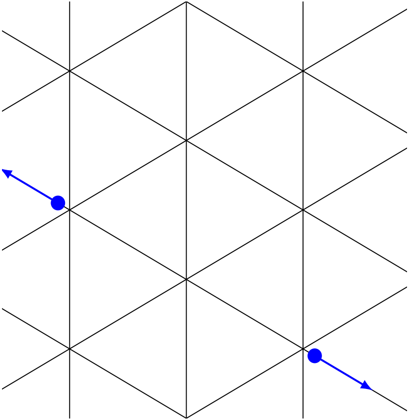
- Regular arrangement of nodes
  - HPP: square lattice $\rightarrow$ does not lead to Navier-Stokes
  - FHP: triangular lattice
- 6 (7) cells per node
  - one for each direction
- for each step, do:
  - collision
  - propagation



after collision

Lattice Gas Automata

- Regular arrangement of nodes
  - HPP: square lattice $\rightarrow$ does not lead to Navier-Stokes
  - FHP: triangular lattice
- 6 (7) cells per node
  - one for each direction
- for each step, do:
  - collision
  - propagation



after propagation

Lattice Gas Automata – advantages and disadvantages

advantages	disadvantages
• boolean state – no rounding errors • easy to implement • easy to parallelize • FHP yields Navier-Stokes	• statistical noise • lack of Galilean invariance

Lattice Gas Automata – advantages and disadvantages

advantages	disadvantages
• boolean state – no rounding errors	• statistical noise
• easy to implement	• lack of Galilean invariance
• easy to parallelize	
• FHP yields Navier-Stokes	

Lattice Gas Automata – advantages and disadvantages

advantages	disadvantages
• boolean state – no rounding errors • easy to implement • easy to parallelize • FHP yields Navier-Stokes	• statistical noise • lack of Galilean invariance

Lattice Gas Automata – advantages and disadvantages

advantages	disadvantages
• boolean state – no rounding errors • easy to implement • easy to parallelize • FHP yields Navier-Stokes	• statistical noise • lack of Galilean invariance

Lattice Gas Automata – advantages and disadvantages

advantages	disadvantages
• boolean state – no rounding errors • easy to implement • easy to parallelize • FHP yields Navier-Stokes	• statistical noise • lack of Galilean invariance

Lattice Gas Automata – advantages and disadvantages

advantages	disadvantages
• boolean state – no rounding errors • easy to implement • easy to parallelize • FHP yields Navier-Stokes	• statistical noise • lack of Galilean invariance

Lattice Gas Automata – advantages and disadvantages

advantages	disadvantages
• boolean state – no rounding errors • easy to implement • easy to parallelize • FHP yields Navier-Stokes	• statistical noise • lack of Galilean invariance

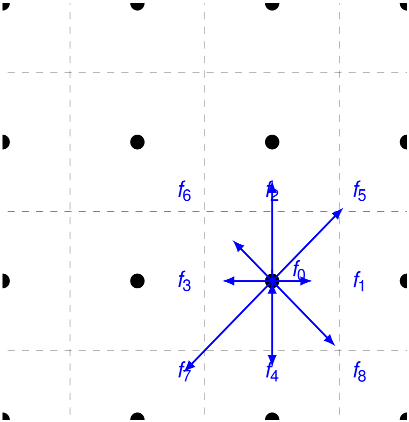
The Lattice Boltzmann Method

Collide Step

Particle distribution function f_i:
discrete in space and time,
continuous in probability values

Collide Step

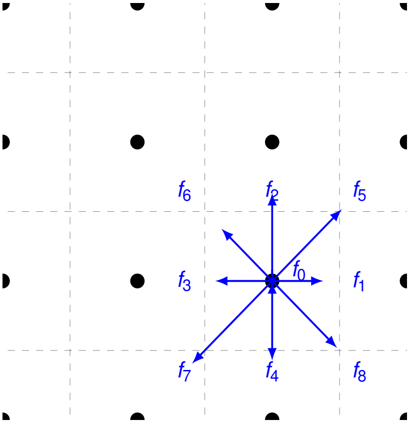
Particle distribution function f_i:
discrete in space and time,
continuous in probability values



One collide step for a D2Q9-stencil

Collide Step

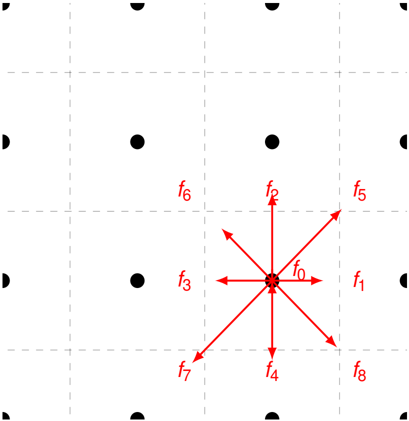
Particle distribution function f_i:
discrete in space and time,
continuous in probability values



One collide step for a D2Q9-stencil

Stream Step

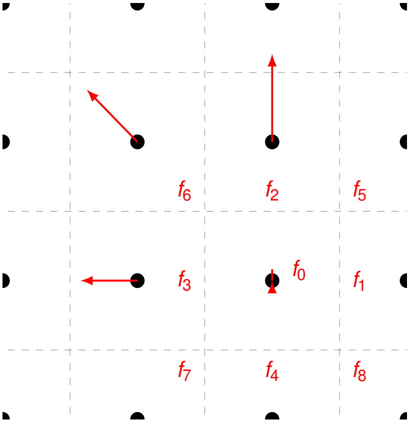
Particle distribution function f_i:
discrete in space and time,
continuous in probability values



One stream step for a D2Q9-stencil

Stream Step

Particle distribution function f_i:
discrete in space and time,
continuous in probability values



One stream step for a D2Q9-stencil

The Lattice Boltzmann Equation

...is what we get when we apply both steps:

$$f_i(\boldsymbol{x} + \boldsymbol{c_i}\Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) + \Omega_i(\boldsymbol{x}, t)$$

The Lattice Boltzmann Equation

...is what we get when we apply both steps:

$$f_i(\boldsymbol{x} + \boldsymbol{c_i}\Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) + \Omega_i(\boldsymbol{x}, t)$$

With collision operator $\Omega_i(\boldsymbol{x}, t)$, e.g.:

- BGK: $\Omega(f) = \frac{1}{\tau}(f_i^{\text{eq}} - f_i) = \omega(f_i^{\text{eq}} - f_i)$
- TRT: is often nicer but needs more formulas $\rightarrow$ please look up

The Lattice Boltzmann Equation

...is what we get when we apply both steps:

$$f_i(\boldsymbol{x} + \boldsymbol{c_i}\Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) + \Omega_i(\boldsymbol{x}, t)$$

With collision operator $\Omega_i(\boldsymbol{x}, t)$, e.g.:

- BGK: $\Omega(f) = \frac{1}{\tau}(f_i^{\mathrm{eq}} - f_i) = \omega(f_i^{\mathrm{eq}} - f_i)$
- TRT: is often nicer but needs more formulas $\rightarrow$ please look up

and the equilibrium particle distribution function $f_i^{\mathrm{eq}} = w_i \rho \left(1 + \frac{\vec{e}_i \vec{u}}{c_s^2} + \frac{(\vec{e}_i \vec{u})^2}{2c_s^4} - \frac{\vec{u}^2}{2c_s^2}\right)$

The Lattice Boltzmann Equation

...is what we get when we apply both steps:

$$f_i(\boldsymbol{x}+\boldsymbol{c_i}\Delta t, t+\Delta t) = f_i(\boldsymbol{x}, t) + \Omega_i(\boldsymbol{x}, t)$$

With collision operator $\Omega_i(\boldsymbol{x}, t)$, e.g.:

- BGK: $\Omega(f) = \frac{1}{\tau}(f_i^{\mathrm{eq}} - f_i) = \omega(f_i^{\mathrm{eq}} - f_i)$
- TRT: is often nicer but needs more formulas $\rightarrow$ please look up

and the equilibrium particle distribution function $f_i^{\mathrm{eq}} = w_i \rho \left(1 + \frac{\vec{e}_i \vec{u}}{c_s^2} + \frac{(\vec{e}_i \vec{u})^2}{2c_s^4} - \frac{\vec{u}^2}{2c_s^2}\right)$

$w_0 = \frac{4}{9}, w_{1,2,3,4} = \frac{1}{9}, w_{5,6,7,8} = \frac{1}{36}$ for D2Q9

The Lattice Boltzmann Equation

...is what we get when we apply both steps:

$$f_i(\boldsymbol{x} + \boldsymbol{c_i}\Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) + \Omega_i(\boldsymbol{x}, t)$$

Cf. Boltzmann Equation

$$\frac{\partial f}{\partial t} + \frac{\boldsymbol{p}}{m} \cdot \nabla f = \left(\frac{\partial f}{\partial t}\right)_{coll}$$

Moments - the Interesting Outputs in LBM

Density and velocity are given locally

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

$$\rho \boldsymbol{u}(\boldsymbol{x}, t) = \sum_i \boldsymbol{c_i} f_i(\boldsymbol{x}, t)$$

Nondimensionalization – setting the step size

= Normalization in the three physical dimensions time, space, mass

Usual choice:

$$\Delta x^* = 1 = C_x \cdot \Delta x$$
$$\Delta t^* = 1 = C_t \cdot \Delta t$$
$$\Delta \rho^* = 1 = C_\rho \cdot \Delta \rho$$

The kinematic velocity in lattice units is directly linked to ω or τ:

$$\nu^* = (\tau - 0.5)c_s^2 = (\frac{1}{\omega} - 0.5)c_s^2, \;\; c_s^2 = \frac{1}{3}$$

$\Rightarrow 0.5 < \tau$ ... and not much larger than 1!

Nondimensionalization – setting the step size

= Normalization in the three physical dimensions time, space, mass

Usual choice:

$$\Delta x^* = 1 = C_x \cdot \Delta x$$
$$\Delta t^* = 1 = C_t \cdot \Delta t$$
$$\Delta \rho^* = 1 = C_\rho \cdot \Delta \rho$$

The kinematic velocity in lattice units is directly linked to ω or τ:

$$\nu^* = (\tau - 0.5)c_s^2 = (\frac{1}{\omega} - 0.5)c_s^2, \;\; c_s^2 = \frac{1}{3}$$

$\Rightarrow 0.5 < \tau$ ... and not much larger than 1!

Nondimensionalization – setting the step size

= Normalization in the three physical dimensions time, space, mass

Usual choice:

$$\Delta x^* = 1 = C_x \cdot \Delta x$$
$$\Delta t^* = 1 = C_t \cdot \Delta t$$
$$\Delta \rho^* = 1 = C_\rho \cdot \Delta \rho$$

The kinematic velocity in lattice units is directly linked to ω or τ:

$$\nu^* = (\tau - 0.5)c_s^2 = (\frac{1}{\omega} - 0.5)c_s^2, \;\; c_s^2 = \frac{1}{3}$$

$\Rightarrow 0.5 < \tau$ ... and not much larger than 1!

Nondimensionalization – setting the step size

= Normalization in the three physical dimensions time, space, mass

Usual choice:

$$\Delta x^* = 1 = C_x \cdot \Delta x$$
$$\Delta t^* = 1 = C_t \cdot \Delta t$$
$$\Delta \rho^* = 1 = C_\rho \cdot \Delta \rho$$

The kinematic velocity in lattice units is directly linked to ω or τ:

$$\nu^* = (\tau - 0.5)c_s^2 = (\frac{1}{\omega} - 0.5)c_s^2, \quad c_s^2 = \frac{1}{3}$$

$\Rightarrow 0.5 < \tau$ ... and not much larger than 1!

Nondimensionalization – setting the step size

= Normalization in the three physical dimensions time, space, mass

Usual choice:

$$\Delta x^* = 1 = C_x \cdot \Delta x$$
$$\Delta t^* = 1 = C_t \cdot \Delta t$$
$$\Delta \rho^* = 1 = C_\rho \cdot \Delta \rho$$

The kinematic velocity in lattice units is directly linked to ω or τ:

$$\nu^* = (\tau - 0.5)c_s^2 = (\frac{1}{\omega} - 0.5)c_s^2, \;\; c_s^2 = \frac{1}{3}$$

$\Rightarrow 0.5 < \tau$ ... and not much larger than 1!

LBM – advantages and disadvantages

advantages	disadvantages
• (moderately) easy to implement • easy to parallelize • yields Navier-Stokes (by Chapman-Enskog analysis) • Galilean invariance	• needs parameter fine-tuning • explicit scheme $\rightarrow$ may need fine time resolution for stability, compared to other CFD methods

LBM – advantages and disadvantages

advantages	disadvantages
• (moderately) easy to implement • easy to parallelize • yields Navier-Stokes (by Chapman-Enskog analysis) • Galilean invariance	• needs parameter fine-tuning • explicit scheme $\rightarrow$ may need fine time resolution for stability, compared to other CFD methods

LBM – advantages and disadvantages

advantages	disadvantages
• (moderately) easy to implement • easy to parallelize • yields Navier-Stokes (by Chapman-Enskog analysis) • Galilean invariance	• needs parameter fine-tuning • explicit scheme $\rightarrow$ may need fine time resolution for stability, compared to other CFD methods

LBM – advantages and disadvantages

advantages	disadvantages
• (moderately) easy to implement • easy to parallelize • yields Navier-Stokes (by Chapman-Enskog analysis) • Galilean invariance	• needs parameter fine-tuning • explicit scheme $\rightarrow$ may need fine time resolution for stability, compared to other CFD methods

LBM – advantages and disadvantages

advantages	disadvantages
• (moderately) easy to implement • easy to parallelize • yields Navier-Stokes (by Chapman-Enskog analysis) • Galilean invariance	• needs parameter fine-tuning • explicit scheme $\rightarrow$ may need fine time resolution for stability, compared to other CFD methods

LBM – advantages and disadvantages

advantages	disadvantages
• (moderately) easy to implement • easy to parallelize • yields Navier-Stokes (by Chapman-Enskog analysis) • Galilean invariance	• needs parameter fine-tuning • explicit scheme $\rightarrow$ may need fine time resolution for stability, compared to other CFD methods

LBM – advantages and disadvantages

advantages	disadvantages
• (moderately) easy to implement • easy to parallelize • yields Navier-Stokes (by Chapman-Enskog analysis) • Galilean invariance	• needs parameter fine-tuning • explicit scheme $\rightarrow$ may need fine time resolution for stability, compared to other CFD methods

Thanks for listening.

Any questions?

References

References

[1] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, *The lattice boltzmann method: Principles and practice*. Springer, Dec. 11, 2016, 705 pp., ISBN: 978-3-319-44649-3.