A brief introduction to the (standard) Lattice-Boltzmann Method and the HONEI approach

Markus Geveler Dirk Ribbrock

Fakultät für Mathematik
Fakultät für Informatik
TU Dortmund

April 9, 2009

Introduction

Overview

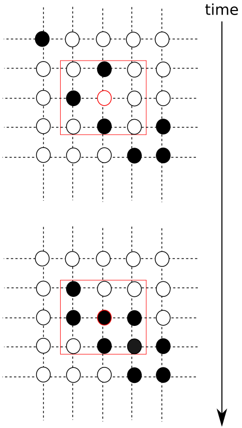
- Overview
- Part one: Introduction to LBM research
  - LBM basics
  - Solving flow problems
  - Typical algorithm
- Part two: Implementation and the HONEI approach
  - *What is this bee again?* - Introduction to HONEI
  - The HONEI LBM framework
  - Optimisations
  - First results
- Conclusion

Overview

Part I: *Introduction to LBM research*

Cellular automata

- regular arrangements of cells of the same kind
- each cell has a finite number of states
- cells are updated simultanously by deterministic rules
- *only a local neighbourhood is taken into account*
- example: Conway's *Game of Life* (1970)



Cellular automata

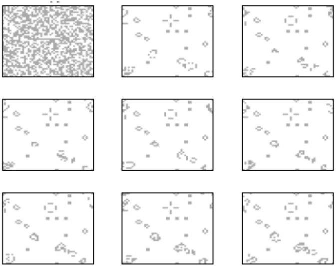
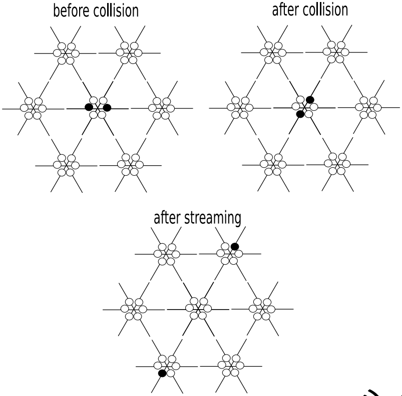


Figure: source: Wolf-Gladrow 2005

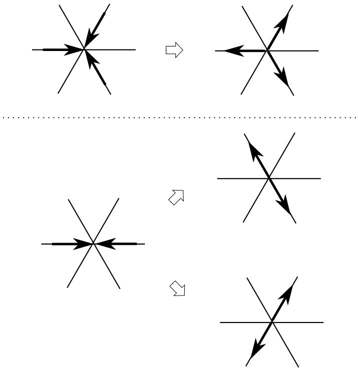
- despite simple update rules: complex behaviour
- easy to implement
- *well suited for massively parallel computers (local updating)*
- but:
    - conservative?
    - propagation of physical quantities?

From CAs to Lattice Gas Cellular Automata

- idea: split update into two steps
- *collision* of particles, which reflects the microdynamics of molecules
- *streaming* of particles over predefined trajectories given by the *lattice*



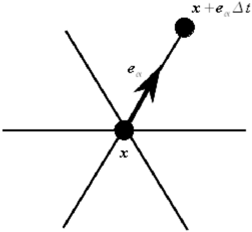
Lattice Gas Cellular Automata



- still simple update rules
- propagation of quantities
- streaming is deterministic but with *statistical noise* $\Rightarrow$ averaging is needed

From LGCAs to LBM: The 'Boltzmann' in Lattice-Boltzmann

In LGCAs, the particle balance equation for particle N in direction α can be written as



$$N_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = N_\alpha(\mathbf{x}, t) + \Omega_\alpha[N(\mathbf{x}, t)]$$

Introduction to LBM research

LBM basics

From LGCAs to LBM: The 'Boltzmann' in Lattice-Boltzmann

Expanding the left hand side of the equation gives

$$N_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = N_\alpha(\mathbf{x}, t) + \Omega_\alpha[N(\mathbf{x}, t)]$$

$$\Leftrightarrow N_\alpha(\mathbf{x}, t) + \Delta t \frac{\partial N_\alpha}{\partial t} + \mathbf{e}_\alpha \Delta t \nabla N_\alpha + \mathcal{O}((\Delta t)^2) = N_\alpha(\mathbf{x}, t) + \Omega_\alpha[N(\mathbf{x}, t)]$$

$$\Leftrightarrow \frac{\partial N_\alpha}{\partial t} + \mathbf{e}_\alpha \nabla N_\alpha = \frac{1}{\Delta t} \Omega_\alpha[N(\mathbf{x}, t)]$$

Which is similar to the Boltzmann equation

$$\underbrace{\frac{\partial f}{\partial t} + \mathbf{v} \nabla f}_{\text{transport term}} = \underbrace{Q}_{\text{collision integral}}$$

From LGCAs to LBM: The 'Boltzmann' in Lattice-Boltzmann

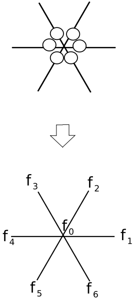
The final step to LBM: replace the boolean fields by *distribution functions* to get rid of the statistical noise

$$N_\alpha(\mathbf{x}+\mathbf{e}_\alpha \Delta t, t+\Delta t) = N_\alpha(\mathbf{x}, t) + \Omega_\alpha[N(\mathbf{x}, t)]$$

$$\rightarrow f_\alpha(\mathbf{x}+\mathbf{e}_\alpha \Delta t, t+\Delta t) = f_\alpha(\mathbf{x}, t) + \Omega_\alpha[f(\mathbf{x}, t)]$$

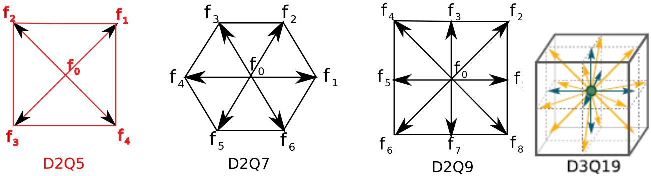
$$\Leftrightarrow$$

$$\underbrace{f_\alpha(\mathbf{x}+\mathbf{e}_\alpha \Delta t, t+\Delta t) - f_\alpha(\mathbf{x}, t) = \Omega_\alpha[f(\mathbf{x}, t)]}_{\text{Lattice Boltzmann Equation}}$$



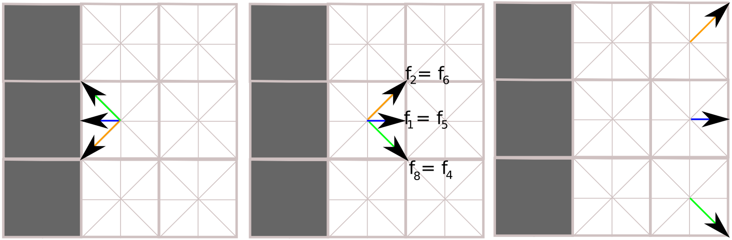
$\Rightarrow$ Still yielding all advantages of LGCA

DxQy what? - The 'Lattice' in Lattice-Boltzmann



unable to recover Navier Stokes

most common lattice type



The collision operator: LBGK models

- in general, Ω_α is a matrix describing the microscopic dynamics
- until early 1990s: non-linear collision operator (evaluation demanding)
- later: linearisation and substitution by the *single relaxation time* model after the collision model by Bhatnagar, Gross, Krook

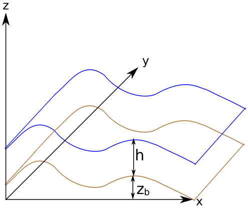
$$\Omega_\alpha = -\frac{1}{\tau}(f_\alpha - f_\alpha^{\mathrm{eq}})$$

which leads to the most common form of the Lattice-Boltzmann equation

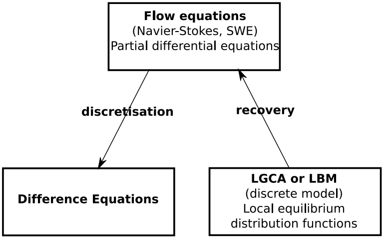
$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) - f_\alpha(\mathbf{x}, t) = -\frac{1}{\tau}(f_\alpha - f_\alpha^{\mathrm{eq}})$$

Local Equilibrium distribution functions: SWE for D2Q9

$$\mathbf{U}_t + F(\mathbf{U})_x + G(\mathbf{U})_y = S(\mathbf{U}),$$
$$(x, y) \in \Omega, t \geq 0,$$
$$\mathbf{U} = \begin{pmatrix} h & hu_1 & hu_2 \end{pmatrix}^T$$

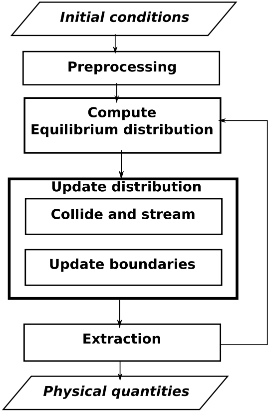


Local Equilibrium distribution functions: SWE for D2Q9



$$f_\alpha^{\mathrm{eq}} = \begin{cases} h - \frac{5gh^2}{6e^2} - \frac{2h}{3e^2} u_i u_i & \alpha = 0 \\ \frac{gh^2}{6e^2} + \frac{h}{3e^2} e_{\alpha i} u_i + \frac{h}{2e^4} e_{\alpha j} u_i u_j - \frac{h}{6e^2} u_i u_i & \alpha = 1,3,5,7 \\ \frac{gh^2}{24e^2} + \frac{h}{12e^2} e_{\alpha i} u_i + \frac{h}{8e^4} e_{\alpha j} u_i u_j - \frac{h}{24e^2} u_i u_i & \alpha = 2,4,6,8 \end{cases}$$

Bringing everything together



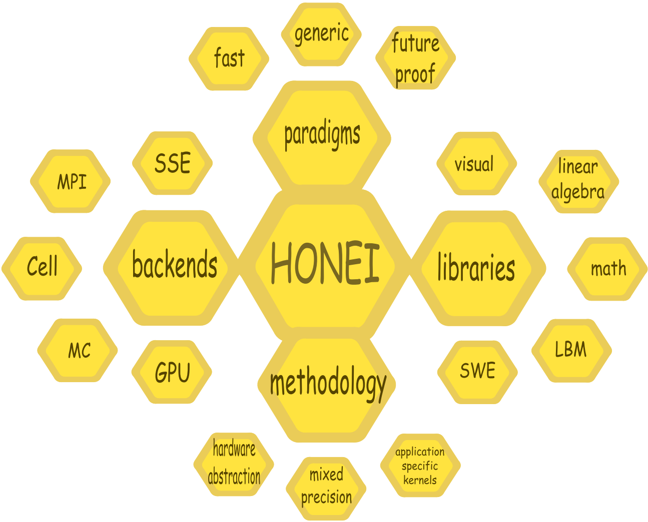
Extract physical quantities:

$$h(\mathbf{x},t) = \sum_\alpha f_\alpha(\mathbf{x},t),$$

$$u_i(\mathbf{x},t) = \frac{1}{h(\mathbf{x},t)} \sum_\alpha e_{\alpha i} f_\alpha$$

Part II: *Implementation and the HONEI approach*

Hardware Oriented Numerics Efficiently Implemented - overview



General

Implementations of all modules for all backends + infrastructure

Extensions to the LBM approach for SWE

External forces do not influence collision $\Rightarrow$ add force terms to distribution

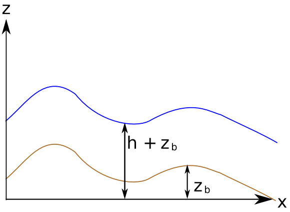
$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) - f_\alpha(\mathbf{x}, t) = -\frac{1}{\tau}(f_\alpha - f_\alpha^{\text{eq}}) + \frac{\Delta t}{6e^2} e_{\alpha i} F_i$$

Numerical evaluation: semi implicit centered scheme (Zhou 2004):

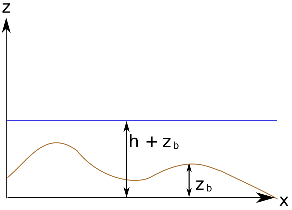
$$F_i = F_i(\mathbf{x} + \tfrac{1}{2} e_{\alpha i} \Delta t, t)$$

Force terms for SWE

Example: still water above uneven bed: *bed slope* $F_i = -gh\frac{\partial z_b}{\partial x_i}$



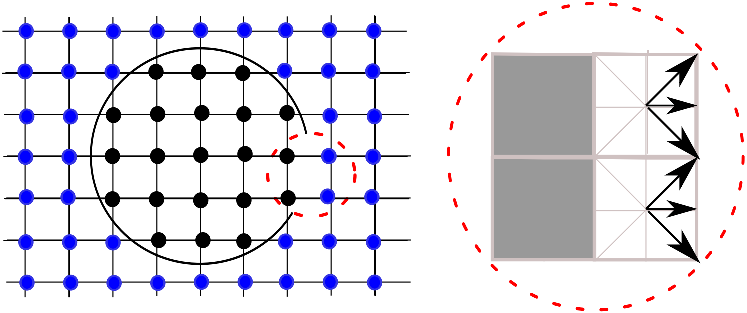
Steady state without force term



Steady state with force term

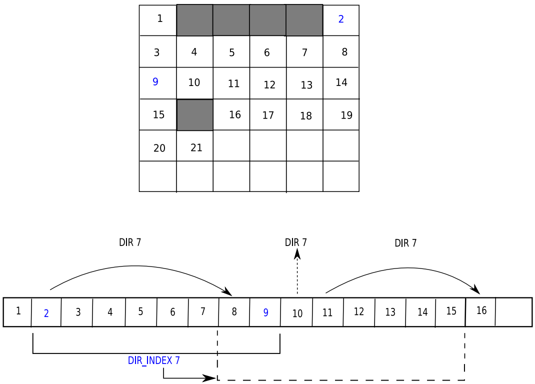
HONEI LBM grid: arbitrary geometry

Rasterization to create obstacle matrix: Preprocessed boundary information



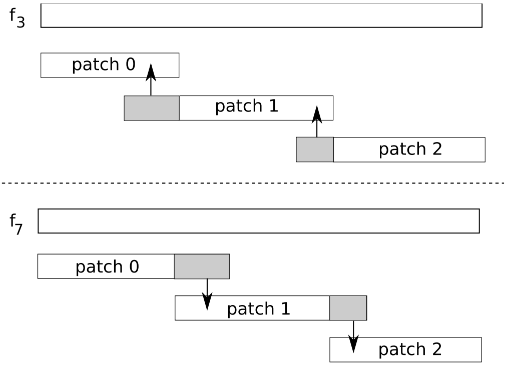
HONEI LBM grid: streaming directions

- no obstacles (solid cells) present in computation
- supports SIMD in all directions



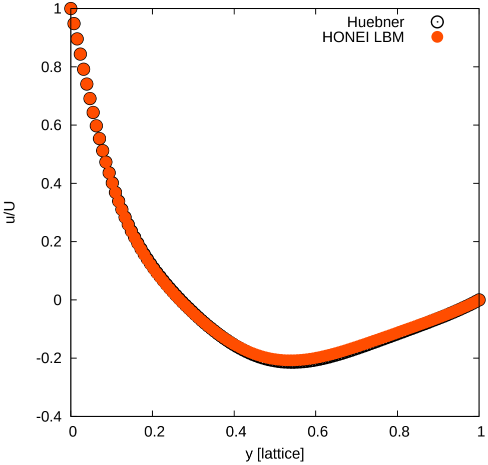
HONEI LBM grid: partitioning

- enables simple 1D partitioning

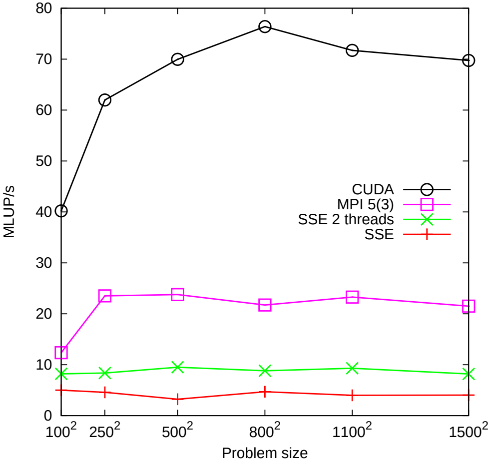


Accuracy

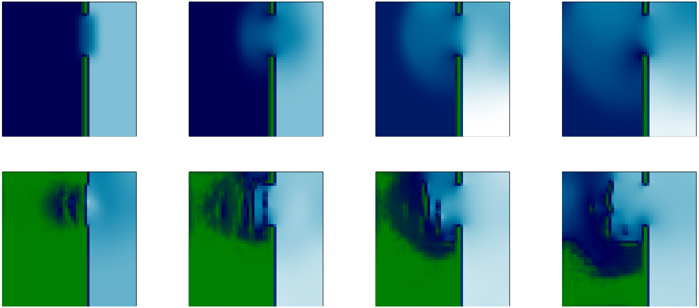
Driven cavity, $R_e = 100$, u_x, mid vertical line, relative, $||u - u_{\text{ref}}||_2 = 0.0002$



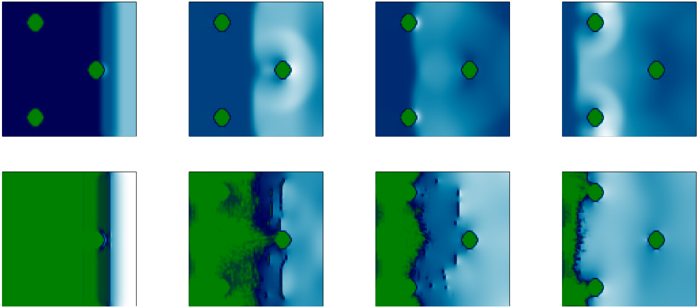
Single precision performance



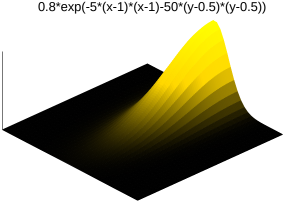
HONEI LBM (SWE) showcases

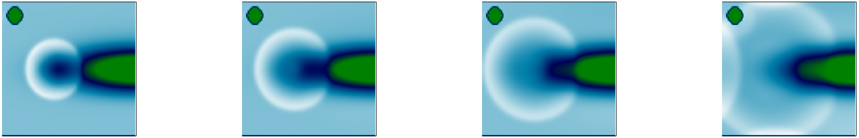


HONEI LBM (SWE) showcases



HONEI LBM (SWE) showcases





Conclusion and future work

The LBM

- strongly related to the most general equation for gas-dynamic flow processes
- efficient
- easy to implement, particularly for complicated boundary schemes
- basic algorithms in 2D quite accurate

Conclusion and future work

The next steps

- provide all modules for all backends
- higher order boundary scheme
- adaptivity
- to the limits of real-time

Thank you for your attention

www.honei.org