Computational Fluid Dynamics with the Lattice Boltzmann Method

Overview, computational issues and (biomedical) applications

Alfons Hoekstra
Computational Science
University of Amsterdam
A.G.Hoekstra@uva.nl
http://www.science.uva.nl/research/scs
This you (probably) know

Fluid velocity

\[ \nabla \cdot \mathbf{u} = 0 \]

\[ \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u} \]

pressure

viscosity

• The Navier-Stokes Equations for an incompressible fluid.
Why an alternative?

CFD solvers are pretty good, aren’t they?
The Boltzmann Equation - 1

\[
\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} = \Omega(f)
\]

\( f(\mathbf{x}, \mathbf{v}) \): single particle distribution function
\( \Omega(f) \): Boltzmann collision term (highly non-linear in \( f \))

Collision term must satisfy conservation of mass, momentum and energy.
Macroscopic fields are obtained by taking velocity moments of $f$.

\[ \rho(x) = \int f(x, v) dv \]
\[ \rho(x)u(x) = \int vf(x, v) dv \]
\[ \rho(x)e(x) = \int \frac{1}{2} |v - u(x)|^2 f(x, v) dv \]

In the limit of low Knudsen and low Mach numbers, it can be shown that these fields obey the Navier-Stokes equations.
The Lattice Boltzmann Method

1. Discretize velocity space in a very small set of velocities;
2. Use a very simple collision operator, usually the BGK collision operator is applied;
3. Discretize space in a lattice with enough symmetry;
4. Use finite differencing for the differential operators
5. Choose the time step, lattice spacing, and discrete set of velocities, such that fit exactly, allowing streaming over the links of the lattice and collisions on the nodes.
Routes to the Lattice Boltzmann Equation (LBE)

- Lattice Gas Automata
- Boltzmann Equation

Lattice Boltzmann Equation

Navier Stokes Equations
The Lattice Gas Cellular Automaton


- Regular lattice with enough symmetry.
- Particles live on the lattice.
  - Streaming over the links;
  - Collisions at the nodes;
    - conservation of mass, momentum, and energy.
- Very easy simulation, trivial for parallel computing.
- One can prove that this LGA recovers the Navier-Stokes equations.
The Hexagonal Lattice

Start

Streaming

Collision
Evolution Equation

\[ n_i(x + c_i, t + 1) = n_i(x, t) + \Delta_i(n(x, t)) \]

- **Streaming**
- **Collision**

**State Vector**

Average occupation number

\[ f_i = \langle n_i \rangle \]

Density

\[ \varphi = \sum_{i=1}^{b} f_i \]

Velocity

\[ \varphi u = \sum_{i=1}^{b} f_i c_i \]
Flow past a Cylinder

single iteration        after averaging
Recover Hydrodynamics

1. Assume molecular chaos
2. Ensemble averaging and Taylor expansion of evolution equation
3. Apply mass and momentum conservation
4. Solve equations using Chapman-Enskog expansion
Ensemble Averaging

• Ensemble averaging of the evolution equation leads to

\[ \langle n_i(x + c_i, t + 1) \rangle = \langle n_i(x, t) \rangle + \langle \Delta_i(n(x, t)) \rangle \]

• which becomes, using the definitions and the molecular chaos assumption

\[ f_i(x + c_i, t + 1) = f_i(x, t) + \Delta_i(f(x, t)) \]

• This is the **Lattice Boltzmann Equation**!
Lots of algebra

\[
\frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial x_\alpha} c_{i,\alpha} \left( \frac{\partial^2 f_i}{\partial x_\alpha \partial x_\beta} c_{i,\beta} \right) + \frac{1}{2} \Delta t \left( \frac{\partial^2 f_i}{\partial t^2} \right) + 2 \frac{\partial f_i}{\partial x_\alpha} c_{i,\alpha} = \frac{1}{\Delta t} \sum_{j=0}^b M_{ij} (f_j - f_j^{eq})
\]

\[
\frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial x_\alpha} c_{i,\alpha} = \frac{1}{\Delta t} \sum_{j=0}^b M_{ij} f_j^{(1)}
\]

\[
\frac{\partial f_i}{\partial t_2} + \sum_{j=0}^b \left( \delta_{ij} + \frac{1}{2} M_{ij} \right) \left( \frac{\partial f_j}{\partial t_1} + \frac{\partial f_j}{\partial x_\alpha} c_{i,\alpha} \right) = \frac{1}{\Delta t} \sum_{j=0}^b M_{ij} f_i^{(2)}
\]

\[
\sum_{i=0}^b \left[ \frac{\partial f_i}{\partial t_1} + \frac{\partial f_i}{\partial x_\alpha} c_{i,\alpha} \right] = \frac{1}{\Delta t} \sum_{i=0}^b \sum_{j=0}^b M_{ij} f_i^{(1)}
\]

\[
\sum_{i=0}^b c_{i,\alpha} \left[ \frac{\partial f_i}{\partial t_1} + \frac{\partial f_i}{\partial x_\alpha} c_{i,\alpha} \right] = \frac{1}{\Delta t} \sum_{i=0}^b c_i \sum_{j=0}^b M_{ij} f_i^{(1)}
\]

\[
\sum_{i=0}^b c_{i,\alpha} \left[ \frac{\partial f_i}{\partial t_2} + \sum_{j=0}^b \left( \delta_{ij} + \frac{1}{2} M_{ij} \right) \left( \frac{\partial f_j}{\partial t_1} + \frac{\partial f_j}{\partial x_\alpha} c_{i,\alpha} \right) \right] = \frac{1}{\Delta t} \sum_{i=0}^b c_i \sum_{j=0}^b M_{ij} f_i^{(2)}
\]

\[
\Pi_{\alpha\beta} = \sum_{i=0}^b c_{i,\alpha} c_{i,\beta} \left( f_i^{eq} + \sum_{j=0}^b \left( \delta_{ij} + \frac{1}{2} M_{ij} \right) f_j^{(1)} \right)
\]

\[
\Pi_{\alpha\beta}^{(0)} = \alpha_s^2 \delta_{\alpha\beta} + \alpha_q \alpha_u \delta_{\alpha\beta}
\]

\[
\Pi_{\alpha\beta}^{(1)} = -c_s^2 \Delta_t \left( \frac{1}{\alpha} - \frac{1}{2} (\frac{\partial \rho}{\partial x_\alpha} + \frac{\partial \rho}{\partial x_\beta}) \right)
\]
Final Result

• To second order we find

$$\frac{\partial \mathbf{u}_\alpha}{\partial \mathbf{x}_\alpha} = 0$$

$$\frac{\partial \mathbf{u}_\alpha}{\partial t} + \frac{\partial}{\partial \mathbf{x}_\beta} \Pi_{\alpha\beta} = 0$$

$$\Pi^{(0)}_{\alpha\beta} = \mathcal{K}^2 \delta_{\alpha\beta} + \mu \mathbf{u}_\alpha \mathbf{u}_\beta$$

$$\Pi^{(1)}_{\alpha\beta} = c_s^2 \Delta t \left( \frac{1}{\lambda} - \frac{1}{2} \right) \left( \frac{\partial \mathbf{u}_\alpha}{\partial \mathbf{x}_\beta} + \frac{\partial \mathbf{u}_\beta}{\partial \mathbf{x}_\alpha} \right)$$

• i.e., an incompressible fluid, with kinematic viscosity

$$\nu = \frac{c_s^2 \Delta t}{\lambda - \frac{1}{2}}$$

• Where $\lambda$ is an eigenvalue of the linearized collision matrix
Properties of LGA

• Advantages
  – local interactions, i.e. inherent parallelism
  – binary operations, i.e. unconditionally stable

• Disadvantages
  – lack of Galilean invariance
  – noisy dynamics
    • note however that the noise contains a lot of interesting physics….
Lattice Boltzmann Method

• Apply directly the densities $f_i$.
• Use very simple collision operator
  – Lattice-BGK
• No need for averaging!
• Galilean Invariant by construction!
Example, the L-BGK D2Q9 model

- 2D square Lattice
- lattice spacing $\Delta x$ and timestep $\Delta t$
- 9 discrete velocities
- BGK operator; single time relaxation towards equilibrium

\[
\Delta(f_i) = \frac{1}{\tau} \left( f_i^{eq} - f_i \right)
\]

\[
\begin{align*}
c_i &= \frac{\Delta x}{\Delta t} e_i, \quad 0 \leq i \leq 9 \\
e_0 &= 0 \\
e_i &= \begin{cases} \cos(\pi/2(i-1)) \\ \sin(\pi/2(i-1)) \end{cases}, \quad i = 1, 3, 5, 7 \\
e_i &= \sqrt{2} \begin{cases} \cos(\pi/2(i-1) + \pi/4) \\ \sin(\pi/2(i-1) + \pi/4) \end{cases}, \quad i = 2, 4, 6, 8
\end{align*}
\]
The L-BGK D2Q9 model

The resulting equations are

\[ f_i(x + e_i \Delta x, t + \Delta t) = f_i(x, t) + \frac{1}{\tau} \left( f_i(x, t) - f_i^{eq}(x, t) \right) \]

\[ f_i^{eq} = \varphi w_i \left[ 1 + 3 e_i \cdot u + \frac{9}{2} (e_i \cdot u)^2 - \frac{3}{2} u^2 \right] \]

\[ w_0 = \frac{4}{9}, \quad w_1 = w_3 = w_5 = w_7 = \frac{1}{9}, \]
\[ w_2 = w_4 = w_6 = w_8 = \frac{1}{36} \]
Recover hydrodynamics from L-BGK

\[ \varrho = \sum_{i=0}^{8} f_i \]
\[ \varrho u = \sum_{i=0}^{8} f_i e_i \]

Within the limits of low Mach and low Knudsen number, the Navier-Stokes equations are recovered, with

\[ p = \varrho / 3 \implies c_s = 1/\sqrt{3} \]
\[ \varrho^* = (2 \tau - 1)/6 \]
3D L-BGK

- 19 discrete velocities
- Same equations as in 2D
- D3Q19 L-BGK has been our workhorse for many years.
Permeability in a Random Fiber Web

Image Based CFD

1. Segmentation of medical images
2. VR based system to change the images …
   • to add a bypass (*virtual surgical procedure*)
   … and to prepare the L-BGK lattice.
3. Image based CFD
   • For blood flow simulation
   • We use the Lattice Boltzmann Method
4. Visualisation kernels
5. A interactive problem solving environment to glue all this together
   • A grid based PSE

Ramos AT, Sloot PMA, Hoekstra AG, Bubak MT. 2004. An Integrative Approach to High-Performance Biomedical Problem Solving Environments on the Grid. Parallel Computing 30:1037—55

The problem solving environment
D-VRE

Step 4: create computational mesh
A few results: flow in the lower abdominal aorta
A few results: flow in an Abdominal Aorta Aneurysm
systolic flow in the superior mesenteric artery

Simulation of systolic flow in the superior mesenteric artery. The arterial structure (a); a snapshot of the simulation (b); a comparison of the velocity profiles between LBM (bullets) and FEM (solid lines) at the region and B at 0.04 second intervals throughout one systolic period, with velocities presented in m/s. (structure and FEM data from University of Sheffield, UK)
Carotid artery with stenosis
Parallelism

• Inherent local model \(\Rightarrow\) trivial parallelism

• Non-homogeneous domain
  – which decomposition?

• Dynamically changing objects
  – load balancing
    • redundant scattered decomposition
    • dynamic load balancing

Non-Homogeneous domain

- Slice decomposition vs. ORB
Parallel Sparse LBM

• Flow simulations in large and complex geometries are *computationally intensive*
  – e.g. porous media, or geometries from medical applications

• We need to
  – Minimize of computational time
    • ➔ Parallelization
  – Minimize of memory requirements
    • ➔ Sparse implementation
Parallel Sparse LBM

• The storage of density distributions is only for the fluid nodes
• The complete domain is mapped on an unstructured grid
• A sorted index list of fluid nodes is composed, where each node has the index list of its neighboring nodes
• 1D array is stored, where only 2*N*19 Reals are for density distributions and N*18 Integers for the adjacency list
Graph Partitioning

• Keep the number of nodes on the sub graphs as equal as possible
  – to minimize load imbalance)

• Minimize edge-cut
  – reduce communication overheads

• We use multi-level partitioning strategies from the METIS library.
  – Fast
  – High quality partitioning
Partitioning Algorithms

• Multilevel K-way
  – Coarsening only once
  – The coarsest graph is directly partitioned into k partitions
  – The uncoarsening phase is also performed only once.

• Multilevel recursive bisectioning (RB)
  – Perform recursive bisectioning on all phases of the multilevel graph partitioning.
Performance Model

• Performance prediction model.
  – To analyze the parallel scalability
  – To find the sources of loss of parallel efficiency

\[
T_p(N) = \frac{T_1(N)}{p} + \sum_{i} T_i(N)
\]

\[
\xi_p(N) = \frac{T_1(N)}{p \cdot T_p(N)} = \frac{1}{1 + \sum f_i}
\]

\[
f_i = \frac{p \cdot T_i(N)}{T_1(N)}
\]

Fractional Overheads

1. Fractional communication overhead

\[ f_{\text{comm}} = \frac{p \max_j \{2d_j \tau_{\text{setup}} + 2e_j \tau_{\text{send}}\}}{N \times \mathcal{A}(N)} \]

2. Fractional load balance overhead

\[ f_l = \frac{p \max \nmax}{N} - 1 \]

3. Fractional processor speed overhead

\[ f_s = \frac{p \max (\mathcal{A}(\nmax) - \mathcal{A}(N))}{N \times \mathcal{A}(N)} = \frac{p \max}{N} \left( \frac{\mathcal{A}(\nmax)}{\mathcal{A}(N)} - 1 \right) \]

Experiments

3 – different geometries
  • Abdominal aorta (AA)
  • Porous media (PM)
  • Straight square channel (SSC)
3 – data sizes
2 – decomposition functions
2 – PC cluster & NEC SX-8 machine
  • with 1,2,4,16,32,64,128 processors

Total: 36 experiments on each of 1…128 processor

**Single Processor Speed**

- For PC cluster the discontinuity indicates that for the $N < 3 \times 10^3$ data sizes we see the cache effect which can give rise to a super linear speed-up.

- It is well-known that vector machines perform best for large data sizes.

---

**Graphs:**

**Left Graph:**
- Title: Single processor speed on the PC cluster
- X-axis: $\log(N)$
- Y-axis: $\tau_{\text{comp}}$
- Data points indicating cache effect

**Right Graph:**
- Title: Single processor speed on the NEC SX-8
- X-axis: $\log(N)$
- Y-axis: $\tau_{\text{comp}}$
- Data points and model fit

In cache
Partitioning

• The multilevel RB creates more sliced cuts between partitions, while the partitions of multilevel K-way are less structured and have curved cuts.

• From measurements of standard deviation of fluid nodes we hardly see difference between multilevel RB and multilevel K-way for all geometries except for SSC for a very large number of partitions.
**Point to Point Communication**

- The SX-8 vector machines have 8 processors per node, thus we distinguished between on-board and off-board communication times.

- After certain amount of data we see a discontinuity due to the buffer size on both PC cluster and SX-8 machine.

---

**Point-to-point communication on the PC cluster**

**Point-to-point communication on the NEC SX-8**

- on-board model fit
- off-board model fit
- on-board measurements
- off-board measurements
Execution Times

Execution times on the PC cluster for AA

- $5 \times 10^6$ measured
- $5 \times 10^6$ model fit
- $10^5$ measured
- $10^5$ model fit
- $5 \times 10^4$ measured
- $5 \times 10^4$ model fit

Execution times on the NEC SX-8 for AA

- $5 \times 10^6$ measured
- $5 \times 10^6$ model fit
- $10^5$ measured
- $10^5$ model fit
- $5 \times 10^4$ measured
- $5 \times 10^4$ model fit

CPU usage vs. execution time for different data sets and model fits.
Performance measurements

Efficiencies on the PC cluster for AA

For all geometries with $5 \times 10^6$ fl nodes

Communication overheads on the NEC SX-8

Efficiencies on the NEC SX-8 for AA

Communication overheads on the PC cluster

Load Balance Overheads

Fractions of CPUs

$0.48, 1.6, 32, 128$

$0.48, 1.6, 32, 64, 128$

$10^0, 10^1, 10^2$

$5 \times 10^6$ measured

$10^5$ measured

$5 \times 10^5$ measured

$5 \times 10^5$ model fit

$10^5$ model fit

$5 \times 10^4$ model fit
MultiPhysics MultiScale

• We now use LBM as a solver in multiphysics multiscale applications

• Coupling LBM with solvers for
  – Diffusion
  – Reactions
  – Biological processes
  – Fluid-Structure interaction

• Multiscale modelling
  – Fast systolic flows (LBM) to slowly proliferating cells.
Example, in-Stent Restenosis

- stenosis: occlusion of blood vessels
- treated by stent-deployment
- (fast) periodic flow
- (slow) biological responses
- advection-reaction-diffusion
- tissue mechanics
- ...

Image of stent procedure before and after treatment.
Simplified model

- 2D
- three single scale models
  - bulk flow (lumen)
  - smooth muscle cells (tissue)
  - drug diffusion (tissue)
- initial conditions
- scale map
- connection scheme
- details of single scale models and coupling templates
Scale Separation Map

Legend:
- Inputs/outputs to single-scale models
- Coupling between different-scale models
- Data items passed in coupling templates
initial condition

- geometry:
  - 2D vessel: 1.5mm x 1mm
  - square struts, 90mm
  - tunica: 120mm

- initial conditions:
  - deployment + SMC relaxation
**CxA for ISR: Connection Scheme**

Diagram showing connections between Init, BF, SMC, mapper, and DD.
Single Scale Models: bulk flow

- Lattice Boltzmann Method
  $\Delta t = 10^{-5}s$, $\Delta x = 0.01\text{mm}$
- receive: geometry updates
- send: shear stress at boundary
Single Scale Models: SMC growth

- Agent Based Model
- Structural solver + biological rule-set

\[ \Delta t = 1h, \; \Delta x = 6\mu m \]

- Receive: shear stresses, drug concentration
- Send: cell positions and radii
Single Scale Models: drug diffusion

• Finite Difference
• SOR to determine steady state

\[ T = 1\ h, \ \Delta x = 0.01\ mm \]

• receive: geometry information
• send: drug concentrations
BF2SMC

- receive cell positions
- receive flow stresses
- map onto SMC stresses

Mappers

SMC

mapper

BF

mapper

DD
Mappers

- **Init**
- **BF**
- **DD**
- **SMC**
- **mapper**

**DD2SMC**
- receive cell positions
- receive drug concentrations
- map onto SMC conc
CxA for ISR: preliminary results
CxA for ISR: preliminary results

time = 2 days
**CxA for ISR: preliminary results**

*time = 4 days*
CxA for ISR: preliminary results

\[ \text{time} = 6 \text{ days} \]
CxA for ISR: preliminary results

time = 8 days
CxA for ISR: preliminary results

\[ \text{time} = 10 \text{ days} \]
CxA for ISR: preliminary results

time = 12 days
CxA for ISR: preliminary results

time = 14 days
CxA for ISR: preliminary results

time = 16 days
Final Remarks

• LBM is a powerful CFD solver for incompressible flows
  – Multiphase
  – Multicomponent
  – Turbulence
• LBM can also be used to model other phenomena
  – Compressible, thermal, non-newtonian flows
  – Shallow water equations
  – Diffusion
  – Waves
• Computationally well understand
  – Serial, parallel, distributed computing
  – Also on GPU, FPGAs, Cell processor
• Active Community
  – Two dedicated annual conferences (DSFD, ICMMES)
  – Many applications in physics, chemistry, biology, biomedicine
Acknowledgements

• Drona Kandhai and many others
• Antti Koponen
• Abdel Monim Artoli
• Lilit Axner
• Alfons Caiazzo
• Eric Lorenz
• Joost Geerdink
• Members of the LBM development consortium