

Manual for LABOETIE

Maximilien Levesque
(Dated: February 21, 2014)

Part I Theory

I. LATTICE GAS CELLULAR AUTOMATA

Lattice Gas Cellular Automata (LGCA) is the ancestor of the lattice boltzmann theory and is a good starting point. In LGGA models, time and space are considered discrete. At every timestep, particles are only allowed to be situated at lattice nodes, with possible velocities \mathbf{c}_i , $i \in \{1, 2, \dots, b\}$. The set \mathbf{c}_i can be chosen in many ways, although they are restricted by the constraint

$$\mathbf{r}' + \mathbf{c}_i \Delta t = \mathbf{r}'', \quad (1)$$

where \mathbf{r}' and \mathbf{r}'' are neighboring lattice nodes. No two particles can be at the same lattice node with the same velocity. Time evolution of the LGGA consists of two steps:

1. *Propagation*: All particles move in one timestep Δt from their initial lattice position \mathbf{r} to a new position $\mathbf{r}' = \mathbf{r} + \mathbf{c}_i \Delta t$. For convenience we chose $\Delta t = 1$.
2. *Collision*: The particles at all lattice nodes undergo a collision that conserves the total number of particles and the total momentum at each node. The collision rules may or may not be deterministic.

The state of the automaton at time t is completely given by $s_i(\mathbf{r}, t)$, which is equal to 1 (0) if a particle is present (absent) on node \mathbf{r} with velocity \mathbf{c}_i . We also define $n_i(\mathbf{r}, t)$, which is the occupation number of a tagged particle. The VACF is then given by

$$\langle v_x(0) v_x(t) \rangle = \sum_{\mathbf{r}_0} \sum_{i_0} \sum_{\mathbf{r}} \sum_i c_{i_0 x} c_{i x} \langle n_{i_0}(\mathbf{r}_0, 0) n_i(\mathbf{r}, t) \rangle. \quad (2)$$

The initial value of the VACF is

$$\langle v_x^2(0) \rangle = \frac{1}{bV} \sum_{\mathbf{r}} \sum_i c_{i x}^2, \quad (3)$$

where V is the number of lattice nodes. With ρ the average number of particles per site, some average values that will be usefull are

$$\langle s_i(\mathbf{r}, t) \rangle = \frac{\rho}{b}, \quad (4)$$

$$\langle n_i(\mathbf{r}, t) \rangle = \frac{1}{bV}, \quad (5)$$

$$\langle n_{i_0}(\mathbf{r}) \rangle \quad (6)$$

...

II. THE MOMENT PROPAGATION METHOD, AS INTRODUCED FOR LGCA

The moment propagation method was first introduced by Frenkel et al. for an efficient estimation of the velocity autocorrelation function (VACF) within the Lattice Gas Cellular Automata simulation method [1–3]. In conventional

techniques, a single particle is followed along its classical trajectory. An estimate of the VACF, $\langle v_x(0) v_x(t) \rangle$, is then obtained as an average of $v_x(0) v_x(t)$ over different time origins in this trajectory, different particles and different initial conditions. To obtain reasonable statistics with these “brute force” methods, one needs lengthy simulations on large systems. We will now present the method called moment propagation (MP), that is about 1 million times more efficient. The reason for this efficiency is that we are willing to sacrifice the complete history of all particles, that is usually irrelevant information. Instead, we compute directly certain averages of the n -body distribution function, without attempting to compute the function itself.

1. First, let us consider one tagged particle, say a "blue" particle, in a lattice with other, but identical "red" particles. We now make use of the fact that in LGCA a particle loses its identity in a collision. So in a collision with a blue particle involved, it is not possible to tell which one of the postcollisional outgoing particles was the incoming blue particle. We now choose our stochastic collision rules such that any outgoing particle is equally likely to be the blue particle. So instead of one defined path of the particle, there are now a lot of possible paths, each associated with a certain probability. We can compute this probability as the product of the scattering probabilities. The average of $v_x(0) v_x(t)$ of the blue particle is the sum over all paths of $v_x(0) v_x(t)$, each path contribution weighted with the appropriate probability factor. Note that only $v_x(t)$ is “path dependent”.
2. Second, we do not need to know which specific path a particle has followed; as a matter of fact, we are only interested in the probability that the blue particle arrives at time t on site \mathbf{r} , because all the paths going to (\mathbf{r}, t) will result in the same $v_x(t)$. This site probability is clearly constructed as the sum of the probabilities of all paths ending at this site. This makes things much easier, for now we can just transport at every time step this probability to the neighbor sites multiplied by the probability that the tagged particle would go that way. Note that at $t = 0$ the site probability for the “starting” site is equal to one, and for all the other sites equal to zero.
3. Finally, we now think of every particle in the lattice as a starting particle. We can just add the contribution of all particles to the site probability, under the condition that the initial site probability of the starting particles is weighted with their initial velocity $v_x(0)$. Note that this procedure corresponds to propagating the v_x velocity of the particles. The correlation function $\langle v_x(0) v_x(t) \rangle$ is now equal to the sum over all sites of this site probability at time t , multiplied by the average site velocity at that time, i.e., we propagated the first moment of the single particle velocity distribution function. More generally, we can propagate any moment of the single-particle distribution function.

The method is conveniently summarized by the following equations: Let us define the number of particles N as

$$N = \sum_{\mathbf{r}} \sum_i s_i(\mathbf{r}, 0) \quad (7)$$

and the average velocity,

$$\bar{v}_x(\mathbf{r}, t) = \sum_i \left(\frac{s_i(\mathbf{r}, t)}{\sum_j s_j(\mathbf{r}, t)} c_{ix} \right). \quad (8)$$

The sum over all tagged particles of the probability to find one individual tagged particle at site \mathbf{r} weighted with its initial velocity is denoted by $W(\mathbf{r}, t)$. It evolves in time as

$$W(\mathbf{r}, t+1) = \sum_i \left(W(\mathbf{r} - \mathbf{c}_i, t) \frac{s_i(\mathbf{r} - \mathbf{c}_i, t)}{\sum_j s_j(\mathbf{r} - \mathbf{c}_i, t)} \right), \quad \forall \mathbf{r} \quad (9)$$

where the initial value $W(\mathbf{r}, 1)$ is given by

$$W(\mathbf{r}, 1) = \sum_i s_i(\mathbf{r} - \mathbf{c}_i, 0) c_{ix}. \quad (10)$$

The VACF at time t is now simply given by

$$\langle v_x(0) v_x(t) \rangle = \frac{1}{N} \sum_{\mathbf{r}} W(\mathbf{r}, t) \bar{v}_x(\mathbf{r}, t). \quad (11)$$

Note that the averaging is all possible paths of one tagged particle and in turn over all possible particles. Compared to conventional techniques, we have a gain in simulation time by six orders of magnitude.

Theory

Major source (copy paste sometimes): Keijo Mattila, PhD, Implementation Techniques for the Lattice Boltzmann Method. The extraordinary simple dynamics with local update rules was ideally suited for parallel computing. McNamara and Zanetti proposed to replace the boolean variables with real numbers (1988). Their proposal in 1988 has been considered as the introduction of the lattice Boltzmann method (LBM).

III. THE LATTICE BOLTZMANN METHOD (LBM)

The lattice Boltzmann model is a pre-averaged version of a lattice gas. The state of the fluid is specified by the average number of particles at each link, $\langle s_i(\mathbf{r}, t) \rangle$, which we denote $n_i(\mathbf{r}, t)$. The hydrodynamic fields, mass density ρ , momentum density \mathbf{j} and momentum flux density $\mathbf{\Pi}$ are simple moments of the velocity distribution,

$$\rho = \sum_i n_i, \mathbf{j} = \sum_i n_i \mathbf{c}_i, \mathbf{\Pi} = \sum_i n_i \mathbf{c}_i \mathbf{c}_i. \quad (12)$$

Like a lattice gas, the system evolves subject to propagation and collision. In going to the LB equation we make use of Boltzmann's 'molecular chaos' hypothesis, i.e., the assumption that there is no correlation between the velocity of a particle at time t and the velocity of its subsequent collision partners (although, as we shall see later, these correlations can be recovered by introducing fluctuations into the system).

The time evolution of the distribution functions n_i is described by the discretized analogue of the Boltzmann equation,

$$n_i(\mathbf{r} + \mathbf{c}_i, t + 1) = n_i(\mathbf{r}, t) + \Delta_i(\mathbf{r}, t), \quad (13)$$

where Δ_i is the change in n_i due to instantaneous molecular collisions at the lattice nodes. The collision operator Δ_i is subject to similar restrictions as those imposed for a lattice gas. A complete description is given by Ladd [4].

IV. MOMENT-PROPAGATION IN LB

The analogy between lattice gases and the lattice Boltzmann equation extends to the moment propagation method. In a lattice gas, the probability that a particle moves with a velocity \mathbf{c}_i after a collision is given by $s_i(\mathbf{r}, t) / \rho(\mathbf{r}, t)$. For the Boltzmann equation it is $n_i(\mathbf{r}, t) / \rho(\mathbf{r}, t)$. The idea is to make use of this fact and to propagate a quantity $P(\mathbf{r}, t)$, defined as the sum of the probabilities of all trajectories which end at node \mathbf{r} at time t , weighted by the α component of their initial velocities. After one timestep we have

$$P(\mathbf{r}, 1) = \sum_i n_i(\mathbf{r} - \mathbf{c}_i, 0) c_{i\alpha}. \quad (14)$$

The time evolution of $P(\mathbf{r}, t)$ is as follows:

$$P(\mathbf{r}, t + 1) = \sum_i \frac{n_i(\mathbf{r} - \mathbf{c}_i, t) P(\mathbf{r} - \mathbf{c}_i, t)}{\rho(\mathbf{r} - \mathbf{c}_i, t)} \quad (15)$$

and the VACF is obtained by multiplying $P(\mathbf{r}, t)$ by the probability that a particle at node \mathbf{r} is currently moving with a velocity component $c_{i\alpha}$. Averaging over all sites this gives

$$\langle c_{i\alpha}(0) c_{i\alpha}(t) \rangle = \frac{1}{N} \sum_{\mathbf{r}} P(\mathbf{r}, t) u_{\alpha}(\mathbf{r}, t), \quad (16)$$

where N is the total number of particles in the system and $u_{\alpha}(\mathbf{r}, t)$ is the local fluid velocity ($= j_{\alpha}(\mathbf{r}, t) / \rho(\mathbf{r}, t)$).

For the 'pure' lattice Boltzmann model, as described above, this is not very interesting because for a fluid at rest, u_{α} is zero at every lattice site and all the terms which we can calculate from the moment propagation term are also zero. This expresses the fact that, having made the molecular chaos assumption, the motions of the particles become

uncorrelated as soon as they have collided. The Boltzmann collision time for the system, τ_B , is therefore just one time step. The VACF is non-zero before any collision occur, i.e., for $t < 1$. The value at $t = 0$ is given by

$$\langle c_{i\alpha}(0) c_{i\alpha}(0) \rangle = \frac{1}{N} \sum_{\mathbf{r}} \sum_i n_i(\mathbf{r}, 0) c_{i\alpha} c_{i\alpha} = \frac{1}{\rho_0} \langle \Pi_{\alpha\alpha}(\mathbf{r}, 0) \rangle, \quad (17)$$

where ρ_0 is the average density.

If, instead of the pure lattice Boltzmann, we consider the 'fluctuating' Boltzmann equation [5], as opposed to the 'pure' Boltzmann equation, then things are different. In the Boltzmann equation the spontaneous fluctuations present in a real fluid are killed off by the pre-averaging. In the fluctuating Boltzmann equation they are reintroduced in the form of imposed random fluctuations in the stress tensor. These stress fluctuations set up microscopic velocity fields in the fluid, so $u_\alpha(\mathbf{r}, t)$ will no longer be zero and the VACF for times greater than zero will also be non-zero. This 'non-Boltzmann' contribution to the VACF should decay in the correct hydrodynamic manner, i.e., $\phi(t) \sim t^{-1}$, and in turn give rise to the theoretically predicted renormalization of the VACF. The remaining problem concerns the relative magnitudes of the Boltzmann contribution and non-Boltzmann (diverging) contribution to the VACF. For the lattice model which we have used,

$$\langle \Pi(\mathbf{r}) \rangle = c_s^2 \langle \rho(\mathbf{r}) \rangle + \langle \rho(\mathbf{r}, t) u_\alpha(\mathbf{r}, t)^2 \rangle, \quad (18)$$

with the speed of sound, c_s , equal to $1/\sqrt{2}$ (all quantities are given in lattice units where the particle mass, time step and lattice spacing are equal to unity). If we assume that $\rho(\mathbf{r}, t) \approx \rho_0$ (a good approximation at low Mach numbers) this implies that

$$\langle c_{i\alpha}(0) c_{i\alpha}(0) \rangle = \frac{1}{2} + \langle u_\alpha^2 \rangle. \quad (19)$$

Since the local velocity, $u(\mathbf{r}, t)$, is constrained to be significantly less than the speed of sound, the first non-Boltzmann term is much smaller than the Boltzmann term - we would have to wait a very long time to reach the fully renormalized region of the VACF (the same problem occurs if one attempts the calculation in a lattice gas). To circumvent this problem we calculated the VACF of a passive tracer which we introduced into the system. The tracer particles are identical to the lattice Boltzmann fluid except they have a probability $\Delta/\rho(\mathbf{r}, t)$ of remaining at the same lattice site.

V. THE COLLISION TERM

LBM comes from a modified version of the Lattice-Gas Automaton (LGA) for the Navier-Stokes equation by Frisch, Hasslacher, and Pomeau (1986).

$$f_i(\mathbf{r} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{r}, t) + \Omega_i(f_i), \quad (20)$$

where one sees in the right hand side that particles move (advection, stream, whatever the name you want to give it), then collide.

A. Lattice BGK (LBGK)

The one we use in the code. Bhatnagar, Gross, Krooks:

$$\Omega_i(f_i) = -\frac{1}{\tau} (f_i(\mathbf{r}, t) - f_i^{eq}(\mathbf{r}, t)). \quad (21)$$

For incompressible flows in BGK scheme,

$$f_i^{eq}(\rho, \mathbf{u}) = \omega_i \rho \left(1 + 3 \frac{\mathbf{e}_i \cdot \mathbf{u}}{c^2} + \frac{9}{2} \frac{(\mathbf{e}_i \cdot \mathbf{u})^2}{c^4} - \frac{3}{2} \frac{\mathbf{u}^2}{c^2} \right), \quad (22)$$

where the weights ω_i depend upon the Lattice BGK model.

Various lattice BGK models as presented by Qian et al. (1992). They enumerate the models with the notation $D_d Q_q$, where d and q refer to the spatial dimension and number of discrete velocities, respectively. The relation $\theta = c_r^2/3$ is true for each model. Likewise, the models have a common expression for the viscosity: $\nu = \theta(1/\omega - 1/2) \Delta t$.

Model	W_0	W_1	W_2	W_3
D1Q3	2/3	1/6	0	0
D2Q9	4/9	1/9	1/36	0
D3Q15	2/9	1/9	0	1/72
D3Q19	1/3	1/18	1/36	0
D4Q25	1/3	1/36	0	0

Table I: Various lattice BGK models as presented by Qian et al. (1992)

VI. KINETIC BOUNDARY CONDITIONS

In a typical scenario, fluid flow is either confined or diverted by solid boundaries – walls if you like. Let $\mathbf{n}(\mathbf{r})$ denote an outward unit normal at the solid boundary Γ_s , i.e. \mathbf{n} points from the solid to the gas domain. The first and most natural boundary condition is the

A. specular reflection law

(*conditions miroirs* in French):

$$f(\mathbf{r}, R\mathbf{c}, t) = f(\mathbf{r}, \mathbf{c}, t), \quad R\mathbf{c} = \mathbf{c} - 2[\mathbf{c} \cdot \mathbf{n}(\mathbf{r})]\mathbf{n}(\mathbf{r}), \quad \mathbf{r} \in \Gamma_s. \quad (23)$$

Here the tangential velocities of impinging molecules remain unchanged, but the normal components of the velocities are reversed. Since specular reflection implies zero friction in the tangential direction of the boundary, molecules so emitted to the gas do not impose drag on the solid body – there are no shear stresses acting on the wall. In reality solid walls do resist gas flows, and thus there has to be a mechanism responsible for friction at the boundary.

B. Diffuse reflection wall

Maxwell took this into consideration (adding drag) by introducing the diffuse reflection law:

$$f(\mathbf{r}, \mathbf{c}, t) = f^M(\rho_W, \mathbf{u}_W, T_W), \quad \mathbf{c} \cdot \mathbf{n}(\mathbf{r}) > 0, \quad \mathbf{r} \in \Gamma_s. \quad (24)$$

where \mathbf{u}_W and T_W are the local wall velocity and temperature, respectively. The local density ρ_W is chosen carefully so that the wall does not accumulate or disperse molecules on that site. Here the colliding molecules interact strongly with the wall, and the emitted molecules leave the wall in a Maxwellian distribution determined by ρ_W , \mathbf{u}_W , and T_W . Finally, Maxwell proposed that a realistic boundary condition, at least to some degree, is a linear combination of the two conditions above: a portion χ of the incident molecules is specularly reflected and the rest $(1 - \chi)$ interact diffusively with the wall. This combination, with various expressions for the accommodation coefficient χ , is still a very popular boundary condition.

C. Reverse reflection law (bounce back), i.e. no-slip

$$f(\mathbf{r}, -\mathbf{c}, t) = f(\mathbf{r}, \mathbf{c}, t), \quad \mathbf{r} \in \Gamma_s, \quad (25)$$

which implies zero fluid flow velocity at the boundary. It is an experimental fact that, in the hydrodynamic regime, most of the real gases in ordinary conditions effectively stick to a solid wall, i.e. they do not slip or, in other words, the relative velocity between the gas and the wall is zero at the boundary. Of course, fluids do slip to some extent when the Knudsen number for a flow is not negligible anymore. For example in the Maxwell's composite boundary condition, the slip can be adjusted with the accommodation coefficient χ . All this suggests that no-slip is an approximation, on a macroscopic level, to what actually happens at the boundary of a fluid. Unfortunately the distinction between the slip and no-slip boundary conditions is not so clear-cut as it may seem from the above discussion. For a *Plane*

Poiseuille flow, that is for a fluid flow between two parallel plates induced by a pressure gradient, the analytical solution he found for the velocity profile is a parabola. However, this parabola is shifted by a constant amount, proportional to the square of the Knudsen number. That is, although the reverse reflection law gives no microscopic slip at the boundary, it may distort the bulk flow by giving rise to a kind of effective macroscopic slip.

This is the only boundary condition actually implemented in the code.

Halfway bounce back.

D. Bounce forward, i.e. slip

VII. EQUILIBRATION OF CHARGES

The first step here is to solve the Poisson-Nernst-Planck equation to find the space repartition of charges.

VIII. EQUILIBRATION WITHOUT CONSTRAINTS

Once the charges have been equilibrated, they're made to move with the solvent. In this step, no external field is applied, i.e., no electric field, pressure gradient, etc.

IX. EQUILIBRATION WITH CONSTRAINTS

Finally, we apply the constraints to the solutes, i.e., some force fields like pressure gradient, electric fields, magnetic fields ...

X. FOLLOW TRACERS DROPPED INTO FLUX OF SOLVENT AND SOLUTES TO GET OBSERVABLES

Here tracers are dropped in the fluid. They may be charged or not. They evolve in the equilibrated fluid and its solutes. They do not change the potential even if they have a charge, i.e., their concentration tends toward zero. The idea is to follow them in order to extract velocity auto-correlation function, while making them not to change the equilibrium properties of the system. Imagine a very small droplet of radioactive particles, so few they do not change anything to the system, but numerous enough to be followed and make statistics.

Part II

Laboetie

XI. INPUT FILE : LB.IN

The input file should be named **lb.in**. For now, almost everything is in this file. For now, Laboetie does not guess default values.

rho_0 (dp) The solvent density, in lattice units (1.0)

XII. MEDIA FILE : MEDIA.IN

A. Sinusoidal surface

The deformable surface function is

$$w^{\pm}(x) = \pm \left[a \sin \left(\frac{2\pi x}{L} \right) + b \right]. \quad (26)$$

Its maximum is $w_{\max/\min}^{\pm} = \pm(a + b)$

XIII. CODE

Laboetie is written in Fortran 2008. It requires gfortran version 7 or later.

A. Precision of variables

All available precisions in the code are defined in module `mod_precision_kinds.f90`. It uses fortran 2008 `iso_fortran_env` to be non-architecture-dependent. Integers can be

- `i0b = INT8` ! 8 bits integer, from -128 to 127 , ie $2^8 - 1$ values
- `i1b = INT16` ! 16 bits integer, from -32768 to 32767
- `i2b = INT32` ! 32 bits integer, from $-2,147,483,648$ to $2,147,483,647$
- `i4b = INT64` ! 64 bits integer, from $-9,223,372,036,854,775,808$ to $9,223,372,036,854,775,807$

Reals can be

- `sp = REAL32` ! 32 bits real number (often called single precision)
- `dp = REAL64` ! 64 bits real number (often called double precision)
- `qp = REAL128` ! 128 bits real number (often called quad precision)

-
- [1] M. A. van der Hoef and D. Frenkel, *Long-time tails of the velocity autocorrelation function in two- and three-dimensional lattice-gas cellular automata: A test of mode-coupling theory*, Phys. Rev. A **41**, 4277 (1990).
 - [2] D. Frenkel in: *Cellular Automata and Modeling of Complex Physical Systems*, P. Manneville, N. Boccara, G. Y. Vichniac and R. Bidaux (editors), Springer, Berlin, 1989, p. 144. (@Max: PAS LU ENCORE)
 - [3] D. Frenkel and M. H. Ernst, *Simulation of diffusion in a Two-Dimensional Lattice-Gas Cellular Automaton: A Test of Mode-Coupling Theory*, Phys. Rev. Lett. **63**, 2165 (1989).
 - [4] A. J. C. Ladd, *Numerical Simulations of Particulate Suspensions via a Discretized Boltzmann Equation Part I. Theoretical Foundation*, J. Fluid. Mech. 271, 285 (1994).
 - [5] A. J. C. Ladd, *Short-time motion of colloidal particles: Numerical simulation via a fluctuating lattice-Boltzmann equation*, Phys. Rev. Lett. 70, 1339 (1993).