Introduction to Lattice Boltzmann Methods

Dragos B. Chirila

February 3, 2010

1 Fluid Equations and Numerical Methods

As you found from the previous lectures, the dynamics of a simple fluid (no thermal effects or tracers considered for now) is described, in the most general form, by the Navier-Stokes equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \tag{1}$$

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) = -\nabla p + \rho \nu \nabla^2 \vec{u} + \rho \vec{f} \tag{2}$$

where:

- $\vec{u}$ is the fluid velocity
- ρ is the density of the fluid
- p is the pressure
- ν is the kinematic viscosity of the fluid and
- $\vec{f}$ is the acceleration due to external forces acting upon the fluid element

After writing down the initial equations, we may employ a series of order-of-magnitude estimates and manipulations to simplify the equations for the particular system (e.g. ocean, atmosphere, industrial flow problem, etc.). One such example was the low-Rayleigh # convection problem, where it was also possible to analytically solve the equations.

However, such fortunate cases are exceptions rather than the rule. In most practical situations, the system is too complex to be reduced to a system of equations which are analytically tractable, especially if we are asking detailed questions (e.g. will it rain tomorrow in Bremen?) instead of more general questions (e.g. what is the average surface temperature on Earth?).

One way of making further progress in such complex situations is to integrate the model equations numerically, just as you did for the Lorenz system. This approach is not devoid of dangers (as additional issues like accuracy of the computer's floating-point representations come into play). However, it is, quite often, the best we can do.

Perhaps one of the first properties of numerical models that the beginner may realize is their diversity. Essentially, the same problem may be approached from different points of view, leading to different solution algorithms but (hopefully) comparable results.

Essential to every such approach is the way physical space is discretized or, more exactly, on which kind of space sub-division is the numerical integration performed. Our exact *partial differential equations* are then ultimately translated to *algebraic difference equations*, which are then solved locally at each space sub-division. The specific solution algorithms are also themselves adapted to the type of space sub-division (also known as *mesh types*), so one generally assigns a name to the pairs of mesh and algorithm.

In climate sciences, we generally speak therefore of Finite Difference- (FDM), Finite Volume- (FVM) and Finite Element-Models (FEM), with some variations of these. However, additional ("non-traditional") approaches are also continuously proposed, one of which is that of Lattice Boltzmann Models (LBM), which we have adopted for this lecture and accompanying assignment.

LBM has many advantages, one of them that is particularly relevant for our current purpose is *simplicity of coding*. If you look at the application accompanying this lecture (the *SimpleLB*-program), you will notice that it only has several hundreds of lines[1], yet it can simulate some non-trivial systems such as the Rayleigh-Benard convection or the von Karman vortex street (more about these later).

[1]Of course, if this is the first Fortran-90 you are seeing, it may still be intimidating. However, to put things in perspective, please note that typical models in the climate system usually have well-above several tens of thousands (sometimes even hundreds of thousands or millions) lines of code.

To keep things unbiased, we shall also note that LBM has several disadvantages, the most serious of them being the relative "stiffness" of the approach relative to the equations it eventually integrates (which means that some extra effort is required if you want it to solve anything *different* from the Navier-Stokes equations; however, once this effort is done, the algorithm simplicity appears once again).

2 Origins of LBM (Lattice Gas Cellular Automata)

Lattice Boltzmann Methods evolved out of Lattice-Gas Cellular Automata (hereafter LGCA), statistical toy-models (inspired by the kinetic theory of gases, to which Ludwig Boltzmann brought significant contributions), which simulated a gas through particles at discrete points in space (e.g. a Cartesian- or hexagonal-mesh - see Fig. (1)), represented by Boolean variables (which also means that the mass of a particle is fixed to $\mathbf{1}$).

LGCA and LBM are both sub-classes of *Cellular Automata*. Common characteristics for all of these models include:

- set of connected sites (the lattice)
- some state-variables defined at each site (several boolean variables for LGCA or several real variables for LBM, as will be explained in next section)
- an update rule, based on local and neighbor information (for LGCA and LBM, we have a composite update rule, namely *collision* and *streaming*)

Perhaps the most important characteristic of the models was the *discretization of velocity space*, which means that particle velocities were restricted to a finite set of orientations. Let us denote with the Greek letter β the number of discrete directions at each lattice point. At each time-step, the particles move along their corresponding directions, approaching the next mesh-point (the term *lattice* is usually preferred in the LBM/LGCA literature to *mesh*). If more than one of these boolean particles arrive simultaneously at the same lattice point, a *collision* rule is applied, which re-distributes the particles such that the conservation laws (for mass and momentum) are satisfied.

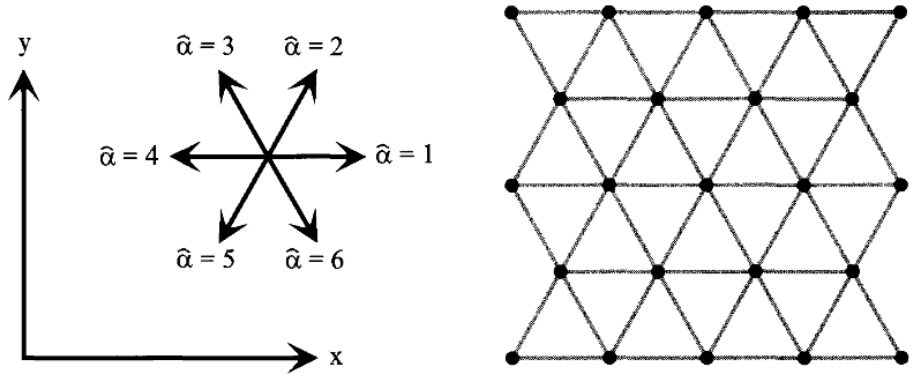

Figure 1: Discretization of space for the triangular LGCA model.

To summarize, the solution algorithm is simply:

1. ***for each*** **time step**

2. ***for each*** **lattice (mesh) point**

3. **Stream()**

4. **Collide()**

Note: Actually, it is not important whether collision is applied before streaming or vice-versa. What matters is only that they are repeated alternatively.

Amazingly enough, these systems exhibited fluid-like behaviour. The prospect of useful fluid simulations became apparent after averaging density and momentum over some regions of space (a procedure also known as *coarse-graining*). An alternative procedure[2] for obtaining the macroscopic fields $(\boldsymbol{\rho}, \vec{\boldsymbol{v}})$ is the method of *ensemble averages*. The idea is simply to consider a set of $\boldsymbol{N}$ (large) simulation set-ups, using the same evolution laws and boundary conditions but different, randomly-assigned, *initial conditions*. Ensemble-averaging the instantaneous (time $\boldsymbol{t}$) occupation numbers in each direction at a specific lattice point ($\vec{\boldsymbol{x}}$), we obtain for that lattice point $\boldsymbol{\beta}$ real numbers $\boldsymbol{f}_{i=1..\beta}$. Each of these numbers $\boldsymbol{f}_i$ represents the probability for site $\vec{\boldsymbol{x}}$ to have a particle heading in the direction $\boldsymbol{i}$, at time $\boldsymbol{t}$. These quantities are called *discretized probability distribution functions* and represent the central link to the Lattice Boltzmann Methods. This is actually a discrete-velocity analogue of the continuous-space

[2]This method is also more rigorous, as it is a well-studied instrument in statistical physics.

single-particle probability distribution functions $\boldsymbol{f(\vec{x}, \vec{v}, t)}$, where:

$$\boldsymbol{f(\vec{x}, \vec{v}, t) d^3x d^3v} \tag{3}$$

represents the probability to find a fluid particle in the volume $\boldsymbol{d^3x}$ and with velocity between $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{v}} + \boldsymbol{d\vec{v}}$.

3 Lattice Boltzmann Methods

Note: We start with the exposition of the simpler, athermal LBM model; later, we will discuss how thermal effects are introduced such that we can simulate the Rayleigh-Benard system.

3.1 Basic LBM

LBM replaces the boolean variables of LGCA [Wolf-Gladrow, 2000] with the discretized probability distribution functions themselves $\boldsymbol{f_i(\vec{x})}$, thereby eliminating the need for ensemble-averaging.

Another difference with respect to LGCA is the simplified collision operator [Bhatnagar et al., 1954].

Also, in contrast to LGCA, there are more choices of the underlying lattice. These are usually classified in the literature using the $\boldsymbol{D\alpha Q\beta}$-notation, where $\boldsymbol{\alpha}$ is an integer number denoting the space dimensionality and $\boldsymbol{\beta}$ is (as introduced before for LGCA) the number of discrete velocities (but now including the possibility of having particle at rest) within the momentum discretization. Some restrictions still have to be fulfilled (especially Galilean and rotational invariance)[3] to ensure that a particular discretization can simulate the Navier-Stokes equations. Among the lattices in common use[4] there are the $\boldsymbol{D2Q9}$ and $\boldsymbol{D3Q19}$-models (see for example discussion in [He and Luo, 1997]). Our focus here is the $\boldsymbol{2D}$ case, hence we have chosen the $\boldsymbol{D2Q9}$ momentum discretization. The discrete velocity directions for the $\boldsymbol{D2Q9}$ lattice are shown in Fig (2).

[3] A lattice with reduced symmetry can be (and has been) used, see [d'Humieres et al., 2001], where a $\boldsymbol{D3Q13}$-lattice is used. However, this approach also departs from the classical BGK-LBM dynamics.

[4] These are the lattices which satisfy these symmetry requirements; as an interesting side-note, the earlier LGCAs failed to recover the Navier-Stokes equations because an improper lattice was used (see [Succi, 2001], pp. 20-21). These models produced square vortices, which were clearly unphysical — one of the reasons why they were regarded as toy-models.

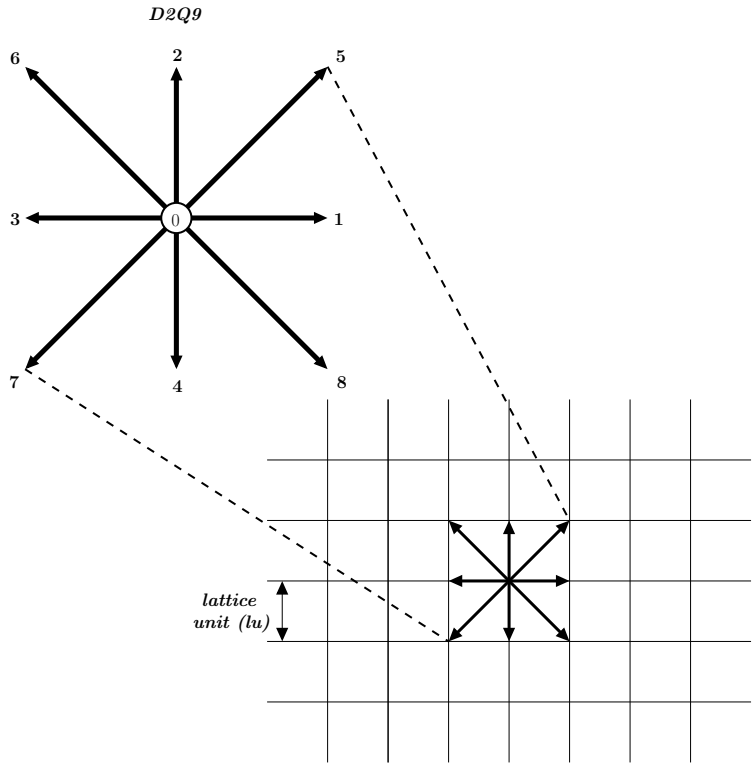

Figure 2: Discrete lattice velocities for the $\boldsymbol{D2Q9}$ model

The macroscopic variables are defined as functions of the particle distribution functions (hereafter DFs) according to:

$$\boldsymbol{\rho} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{\beta-1}} \boldsymbol{f_i} \qquad \text{(macroscopic fluid density)} \tag{4}$$

and

$$\vec{\boldsymbol{u}} = \frac{\mathbf{1}}{\boldsymbol{\rho}} \sum_{\boldsymbol{i=0}}^{\boldsymbol{\beta-1}} \boldsymbol{f_i}\vec{\boldsymbol{e_i}} \qquad \text{(macroscopic velocity).} \tag{5}$$

The DFs at each lattice point are updated using the equation:

$$\boxed{\underbrace{\boldsymbol{f_i}(\vec{\boldsymbol{x}} + \vec{\boldsymbol{e_i}}\boldsymbol{\Delta t}, \boldsymbol{t} + \boldsymbol{\Delta t}) = \boldsymbol{f_i}(\vec{\boldsymbol{x}}, \boldsymbol{t})}_{\text{Streaming}} - \underbrace{\frac{[\boldsymbol{f_i}(\vec{\boldsymbol{x}}, \boldsymbol{t}) - \boldsymbol{f_i^{eq}}(\vec{\boldsymbol{x}}, \boldsymbol{t})]}{\boldsymbol{\tau}}}_{\text{Collision}},} \tag{6}$$

where $\boldsymbol{i} \in [\mathbf{0}, \boldsymbol{\beta} - \mathbf{1}]$ is an index spanning the (discretized) momentum space and $\boldsymbol{\tau}$ is a relaxation parameter, which is related to the fluid viscosity (more details about this will follow in this section).

This equation holds for lattice points within the fluid domain, but not for the domain boundaries, where boundary conditions compensate for the insufficient number of DFs (it does not

make sense to stream DFs from walls towards the fluid). For this reason, the two steps (streaming & collision) are usually treated separately in actual numerical implementations.

The streaming step, where the DFs are translated to the neighbouring sites according to the respective discrete velocity direction, is illustrated in Fig. (3), in the $\boldsymbol{D2Q9}$ model for simplicity. The collision step (illustrated in Fig. [4]) consists of a re-distribution of the DFs towards the local discretized Maxwellian equilibrium DFs, but in such a way that local mass and momentum are invariant[5].

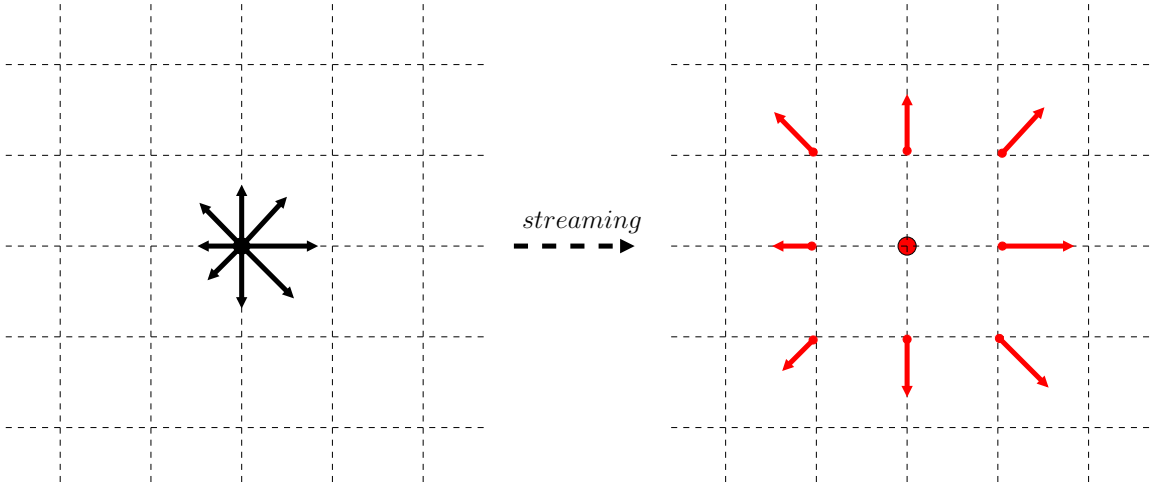

Figure 3: Illustration of the streaming process on a $\boldsymbol{D2Q9}$ lattice. Note that the magnitude of the DFs remain unchanged, but they move to a neighbouring node according to their direction.

The equilibrium DFs can be obtained from the local Maxwell-Boltzmann SPDF (see for example [He and Luo, 1997]); they are

$$\boldsymbol{f_i^{eq}(\vec{x}) = w_i \rho(\vec{x}) \left[1 + 3\frac{\vec{e_i}\cdot\vec{u}}{c^2} + \frac{9}{2}\frac{(\vec{e_i}\cdot\vec{u})^2}{c^4} - \frac{3}{2}\frac{\vec{u}^2}{c^2} \right]}, \tag{7}$$

where for the $\boldsymbol{D2Q9}$ model the weights are

$$\begin{cases} \boldsymbol{w_{i=0}} = \frac{4}{9} \\ \boldsymbol{w_{i=\{1..4\}}} = \frac{1}{9} \\ \boldsymbol{w_{i=\{5..8\}}} = \frac{1}{36} \end{cases} \tag{8}$$

and $\boldsymbol{c}$ is the propagation speed on the lattice $(\boldsymbol{1}\textit{lattice spacing}/\textit{time step})$, taken as $\boldsymbol{c = 1}$ in most cases.

[5]As the method is mostly used in the incompressible limit, energy conservation is equivalent to momentum conservation

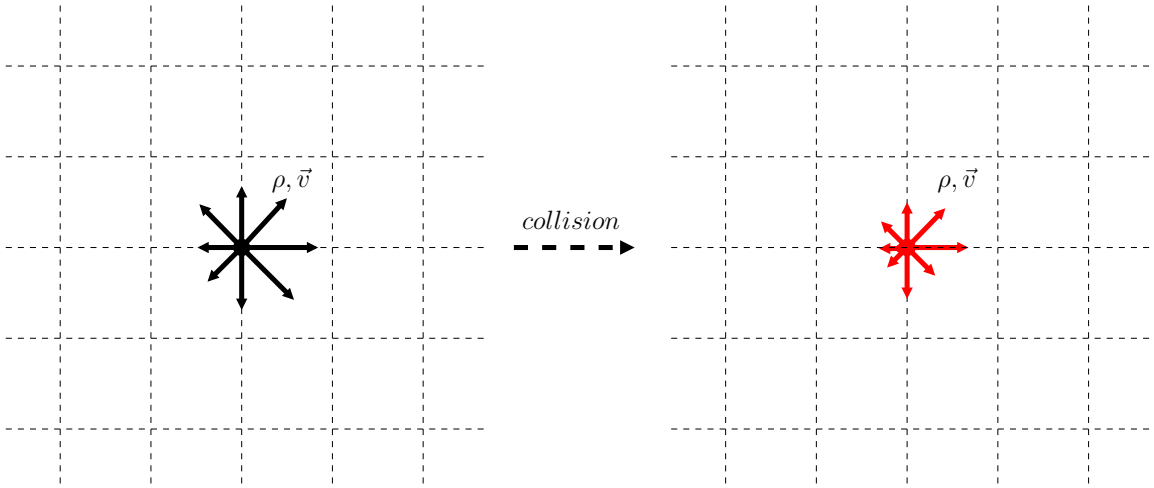

Figure 4: Illustration of the collision process on a $\boldsymbol{D2Q9}$ lattice. Note that the local density $\boldsymbol{\rho}$ and velocity $\boldsymbol{\vec{v}}$ are conserved, but the DFs change according to the relaxation-to-local-Maxwellian rule.

Under the afore-mentioned assumption of a low Mach number, and further taking $\boldsymbol{Kn}$[6], $\boldsymbol{\delta_t}$[7], $\boldsymbol{\delta_x}$[8] $\boldsymbol{\rightarrow 0}$, this model recovers the incompressible Navier-Stokes equations:

$$\boldsymbol{\nabla \cdot \vec{u} = 0}\,, \tag{9}$$

$$\boldsymbol{\rho \partial_t \vec{u} + \rho \vec{u} \nabla \cdot \vec{u} = -\nabla P + \rho \nu \nabla^2 \vec{u}} \tag{10}$$

with an isothermal equation of state:

$$\boldsymbol{P = c_s^2 \rho}\,, \tag{11}$$

where $\boldsymbol{P}$ is the pressure.

The viscosity of the fluid is related to the relaxation parameter $\boldsymbol{\tau}$ by the equation

$$\boldsymbol{\nu = c_s^2 \left(\tau - 1/2\right) \Rightarrow \tau = \frac{\nu}{c_s^2} + \frac{1}{2} \xrightarrow[c_s^2|_{D2Q9}=1/3]{} \tau_{D2Q9} = 3\nu + \frac{1}{2}} \tag{12}$$

The proof of these results follows from the Chapman-Enskog analysis. The speed of sound $\boldsymbol{c_s}$ is a lattice-dependent quantity, which takes the value

$$\boldsymbol{c_s = \frac{1}{\sqrt{3}}}$$

[6]The assumption of $\boldsymbol{Kn \equiv \frac{\lambda}{L} \rightarrow 0}$ is a requirement for continuum models to apply, hence it is not specific to LBM.

[7]Physical time unit, not the computational time unit $\boldsymbol{\Delta t}$, which is usually taken as $\boldsymbol{1}$.

[8]Physical space unit $\boldsymbol{\neq \Delta x}$, which is usually taken as $\boldsymbol{1}$.

for the $D2Q9$ lattice.

Eq. (12) provides a straightforward method for adjusting the fluid viscosity in the model. It is obvious that $\tau \geq 0.5$ is required in order to ensure a positive viscosity. The limit $\tau \longrightarrow 0.5$ corresponds to the inviscid flow, while the $\tau \longrightarrow \infty$ limit represents the Stokes (creeping) flow. While the later case poses no difficulty to the model[9], the former limit is problematic because stability issues appear if an insufficient lattice resolution is chosen. This is due to the fact that velocity gradients can become very large (especially in complex geometries, with high topography variations) and the model cannot dissipate the energy due to the very low viscosity[10].

3.2 Smagorinsky turbulence model

Unfortunately, many of the flows of practical interest are turbulent, often with a low viscosity. To overcome this stability problem, a turbulence model is necessary. The role of this procedure is to parameterize the turbulent energy dissipation in turbulent flows, where the larger eddies extract energy from the mean flow and ultimately transfer some of it to the smaller eddies which, in turn, pass the energy to even smaller eddies, and so on up to the smallest scales, where the eddies convert the kinetic energy into internal energy of the fluid. At this scales (also known as Kolmogorov scale), the viscous friction dominates the flow [Frisch, 1996].

In classical LB applications, a convenient method of modelling turbulent dissipation is through a locally-enhanced collision, which effectively stabilizes the simulation. A common (see [Thuerey, 2007]) procedure is the Smagorinsky sub-grid model [Smagorinsky, 1963], which, when adapted to LBM, consists of:

1. **evaluation of the local stress tensor**:

$$\Pi_{m,n} = \sum_{i=0}^{\beta-1} \vec{e}_{i,m} \vec{e}_{i,n} (f_i - f_i^{eq}) \,, \tag{13}$$

   where $(m, n) \in \{x, y, z\} \times \{x, y, z\}$;

[9] Strictly speaking, the Stokes limit presents its own issues, namely the slow convergence rate; however, in the present context, we concentrate on stability.

[10] The "brute-force" remedy of increasing the grid size improves the situation by effectively rescaling the velocity field, which automatically diminishes velocity gradients. However, this approach quickly becomes unfeasible.

2. **computation of the enhanced relaxation time**:

$$\tau_S = 3(\nu + C^2 S) + \frac{1}{2}\,, \tag{14}$$

$$\text{where: } S \equiv \frac{1}{6C^2}\left(\sqrt{\nu^2 + 18C^2\sqrt{\Pi_{m,n}\Pi_{m,n}}} - \nu\right). \tag{15}$$

Proper values for the Smagorinsky constant C that are suitable for LBM have been published in the literature [Yu et al., 2005] and found to be close to 0.03. It can be observed that, as $S > 0, \forall f_i$, the effect of the model will always be a higher local effective viscosity, which increases as the local stresses increase.

To conclude this discussion, it is worth mentioning that this procedure does indeed complicate the collision operator, losing some of the elegance of the LBM algorithm. However, the efficiency is increased because it allows one to work on much coarser grids compared to the original LBM, at the same value for viscosity [Thuerey, 2007].

3.3 Boundary conditions for athermal LBM

Boundary Conditions (BC) form an important part of any numerical solution. In many cases BCs can strongly influence the accuracy of the algorithm.

Periodic Boundary Conditions (hereafter BCs). The simplest type of boundary condition is the periodic one. In this case, the domain becomes folded along the direction of the periodic BC pair. From the perspective of ocean modelling, this kind of BC is, of course, only useful in preliminary tests, as it implies a high symmetry of the flow domain.

No-slip BCs. The most often used type of BC in LBM flows is the no-slip BC, especially the simple bounce-back rule, which is quite elegant and surprisingly accurate. The basic idea is that the incoming DFs at a wall node are reflected back to the original fluid nodes, but with the direction rotated by π radians.

The bounce-back BC is one of the most advertised benefits of the Lattice Boltzmann method, as it is trivial to implement and it allows one to effortlessly introduce obstacles into the fluid domain (for example, by using a global Boolean field; bounce-back is performed for all of the wall lattice points, whose flag was "flipped" during the initialization stage). However, the BC

has been proven to be only first-order accurate in time and space [Pan et al., 2006]. A straightforward improvement is to consider the wall-fluid interface to be situated halfway between the wall and fluid lattice nodes [Ziegler, 1993]. This simple translation (which is actually nothing more than a slightly different post-processing procedure), commonly referred to as half-way bounce-back in the literature, is illustrated in Fig. (5).

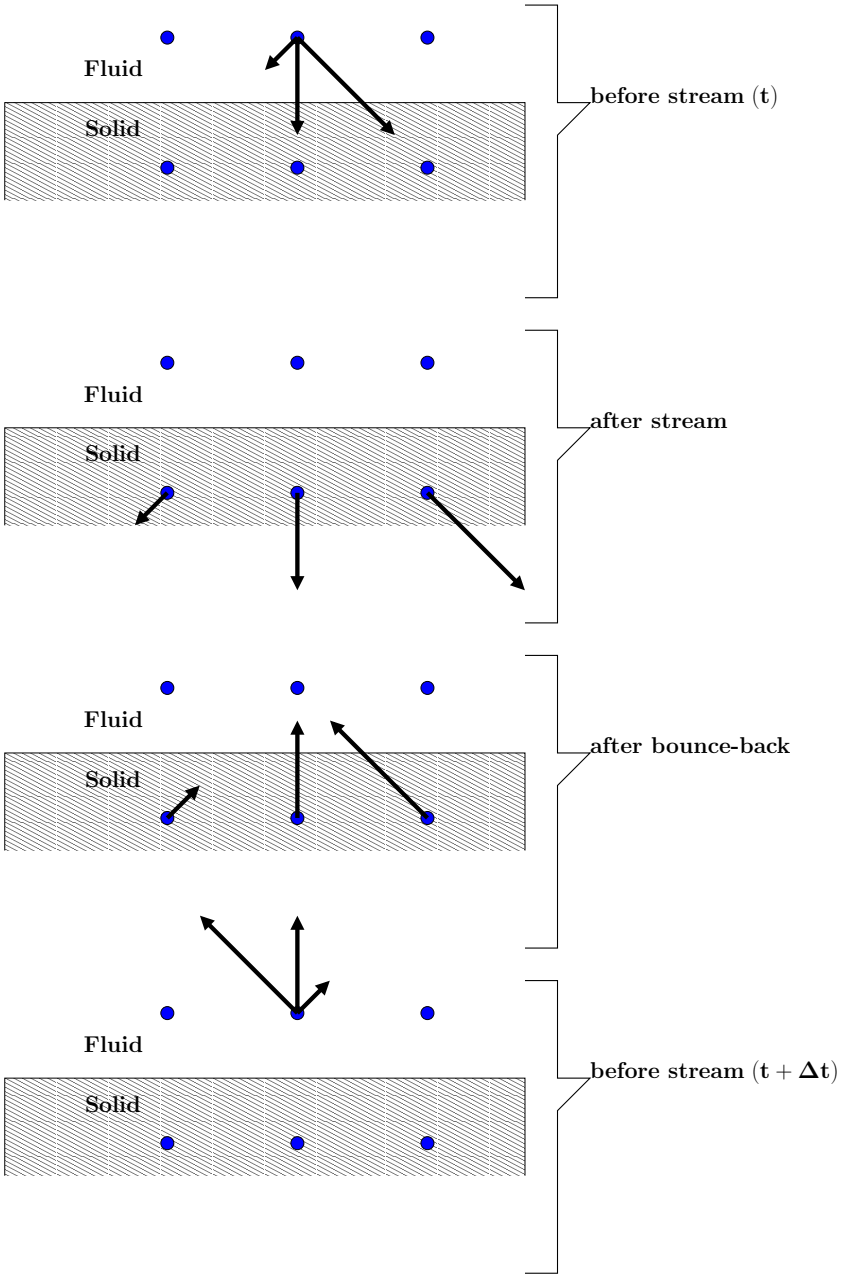

Figure 5: Illustration of the half-way bounce-back algorithm for the $\boldsymbol{D2Q9}$ model (adapted after [Sukop and Thorne, 2006]).

Slip BCs. The slip BC is similar to the no-slip one, except that the DFs are reflected in a mirror-like fashion instead of being bounced-back. A clear treatment of this can be found in [Succi, 2001].

Velocity and Pressure BC. Sometimes, we need to be able to model flows with prescribed velocity (also known as von Neumann BC) or pressure profiles (also known as Dirichlet BC). This is the case, for example, when the simulation domain communicates with other, not simulated but parameterized flow domains. The prescribed velocity or pressure add $\mathbf{2}$ (in $\boldsymbol{D2Q9}$) and, respectively $\mathbf{1}$ equation for determining the unknown DFs (which would hypothetically have to come from "outside" the fluid domain). In the case of the velocity BC, the additional equations are actually enough to solve for the unknown DFs (in $\boldsymbol{D2Q9}$); for the pressure BC, the system of equations is still not closed, and additional constitutive equations are necessary. A central idea in this direction is that of the bounce-back of the non-equilibrium part of the DFs in the normal direction (also known as the Zou-He assumption after its inventors [Zou and He, 1997]). Hence, for a northern boundary, one would write:

$$\boldsymbol{f_2 - f_2^{eq} = f_4 - f_4^{eq}} \tag{16}$$

3.4 Coupled temperature solver for the Boussinesq approximation

The model described so far is only applicable to athermal liquids. While there are many flow situations which can be attributed to this class, thermal effects are often essential to many natural phenomena. Over time, two main extensions of LBM to thermal flows have emerged. The first approach (and the easiest to program) consists of allowing more discrete directions for velocity. By defining new equilibrium DFs and by imposing certain symmetries on the new lattice, an energy equation can be obtained, in addition to the mass and momentum equations. This approach, however, suffers from extreme numerical instability, a limited range of temperature variation and a fixed Prandtl[11] number (because there is still a single relaxation time in the model).

The second approach (and also the one we adopted here) consists of solving the passive

[11]The Prandtl number is a non-dimensional quantity approximating the ratio of momentum diffusivity to that of the thermal diffusivity, $\boldsymbol{Pr = \frac{\nu}{\kappa}}$.

scalar equation for temperature on a separate lattice. The temperature field is influenced by the fluid advection, and influences the fluid through a buoyancy (i.e. force) term. This approach is only valid in the Boussinesq approximation, which is a reasonable assumption for many flows (for example, in ocean flows). Hence, take density variations to be small everywhere, except for the gravity term, where we use the simple equation of state:

$$\boldsymbol{\rho = \rho_0 \left[1 - \alpha(T - T_0)\right]} \tag{17}$$

where $\boldsymbol{\alpha}$ is the coefficient of thermal expansion, $\boldsymbol{T_0}$ is the temperature at which the reference density $\boldsymbol{\rho_0}$ was measured and $\boldsymbol{(\rho, T)}$ are the instantaneous values for density and temperature.

The LB evolution algorithm is the same on the temperature lattice, but with different equilibrium DFs. Also, because the macroscopic quantity is a scalar (in contrast to the LBM for the velocity field, which is a vector), a lattice with fewer velocity directions is sufficient.

Note: To keep things short, we do not describe in detail how the LBM for solving the temperature equation works. Suffice it to say that analogue concepts as for the usual LBM are defined, namely a similar update rule, equilibrium DFs, etc.

References

[Bhatnagar et al., 1954] Bhatnagar, P., Gross, E. P., and Krook, M. K. (1954). A model for collision process in gases. i. small amplitude processes in charged and neutral one-component system. *Phys. Rev*, 94:511.

[Cornubert et al., 1991] Cornubert, R., d'Humieres, D., and Levermore, D. (1991). A knudsen layer theory for lattice gases. *Physica D*, 47:241.

[d'Humieres et al., 2001] d'Humieres, D., Bouzidi, M., and Lallemand, P. (2001). Thirteen-velocity three-dimensional lattice boltzmann model. *PRE*, 63(6, Part 2).

[Frisch, 1996] Frisch, U. (1996). *Turbulence: the legacy of A.N. Kolmogorov*. Cambridge University Press. ISBN 0-521-45103-5.

[He and Luo, 1997] He, X. and Luo, L. S. (1997). Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. *Phys. Rev. E*, 56(6):6811–6817.

[Pan et al., 2006] Pan, C., Luo, L. S., and Miller, C. T. (2006). An evaluation of lattice Boltzmann schemes for porous medium flow simulation. *Comput. Fluids*, 35:898–909.

[Smagorinsky, 1963] Smagorinsky, J. (1963). General circulation experiments with the primitive equations. *Mon. Wea. Rev.*, 91:99–164.

[Succi, 2001] Succi, S. (2001). *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Oxford University Press, Oxford.

[Sukop and Thorne, 2006] Sukop, M. and Thorne, D. T. (2006). *Lattice Boltzmann Modeling: an introduction for geoscientists and engineers*. Springer Verlag, 1 edition. ISBN 978-3-540-27981-5.

[Thuerey, 2007] Thuerey, N. (2007). *Physically based animation of free surface flows with the Lattice Boltzmann Method*. PhD thesis, Der Technischen Fakultaet der Universitaet Erlangen-Nuernberg.

[Wolf-Gladrow, 2000] Wolf-Gladrow, D. (2000). *Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An Introduction*. Springer Verlag, 1 edition. ISBN 978-3-540-66973-9.

[Yu et al., 2005] Yu, H., Girimaji, S., and Luo, L. S. (2005). Lattice Boltzmann simulations of decaying homogeneous isotropic turbulence. *Phys. Rev. E*, 71:016708–1:5.

[Ziegler, 1993] Ziegler, D. P. (1993). Boundary conditions for the lattice Boltzmann simulations. *J. Stat. Phys.*, 71:1171.

[Zou and He, 1997] Zou, Q. and He, X. (1997). On pressure and velocity boundary conditions for the lattice Boltzmann bgk model. *Phys. Fliuds*, 9(6):1591–1598.