4. Forces & collision operators

4.1 Introduction

In this set of lecture notes I would like to discuss a few – largely unrelated – topics that have practical relevance (e.g. for your research).

Exerting local forces on a lattice-Boltzmann fluid requires some special care that we will discuss. This topic is important for many reasons. One is implementing the forces arising from solid particles in an unresolved particles fluid-solids simulation (see LN 5), another is implementing an immersed boundary method to deal with off-lattice no-slip walls.

So far we have been working with the BGK (single relaxation time) collision operator in relation to the equilibrium distribution function. More advanced collision operators are available that have advantages over the BGK operator. We will be discussing the multiple relaxation time (MRT) operator, the way to implement it, and it's possible advantages.

4.2 Incorporating body forces in the lattice-Boltzmann method

The incompressible Navier-Stokes equation as presented in LN01 (Eq. 1.19, repeated here as Eq. 4.1) contained an external force F_α

$$\rho\frac{\partial u_\alpha}{\partial t}+\rho u_\beta\frac{\partial u_\alpha}{\partial x_\beta}=-\frac{\partial p}{\partial x_\alpha}+\frac{\partial}{\partial x_\beta}\left[\mu\left(\frac{\partial u_\alpha}{\partial x_\beta}+\frac{\partial u_\beta}{\partial x_\alpha}\right)\right]+F_\alpha \tag{4.1}$$

It is important to realize that F_α is a *body force*, i.e. a force per unit volume. If F_α would represent gravity it would have the form ρg_α. In the analysis of the lattice-Boltzmann method in LN02 we did not include a body force. Also when discussing the practical aspects in LN03 we did not discuss this topic. We will discuss including body forces here.

There are two main approaches for incorporating a body force. One is via adapting the collision operator. This makes perfect sense since force is a source of momentum and therefore the constraint on the collision operator that it conserves momentum ($\sum_i \Omega_i c_{i\alpha}=0$) should be amended. Another is via including force as a source term in the lattice-Boltzmann equation. For no specific reason we go in these lecture notes with the second approach. Previously (LN02, Eq. 2.10) we wrote $f_i^*=f_i+\Omega_i$ with f_i the pre-collision distribution function, and f_i^* the post-collision distribution function. This expression we now supplement with a source term S_i

$$f_i^*=f_i+\Omega_i+S_i \tag{4.2}$$

The source term S_i relates to the body force F_α in the following manner

$$S_i = \left(1-\frac{1}{2\tau}\right) w_i \left(\frac{c_{i\alpha}}{c_s^2} + \frac{\left(c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta}\right)u_\beta}{c_s^4} \right) F_\alpha \tag{4.3}$$

with the – by now well known – notation for the velocity set (see Eqs. 2.5 & 2.6). Note that Eq. 4.3 applies the summation convention over the two repeated Greek indices α, β.
Equation 4.3 is the result of yet another Chapman-Enskog analysis involving the Navier-Stokes equation that includes a force term. Also as a result of the Chapman-Enskog analysis it has become clear that the velocity requires a *force correction*:

$$\rho u_\alpha = \sum_i f_i c_\alpha + \tfrac{1}{2} F_\alpha \tag{4.4}$$

The density is still calculated according to the familiar expression

$$\rho = \sum_i f_i \tag{4.5}$$

For completeness, we also give the expression for the deviatoric viscous stress that – as you remember – was due to deviations of the distribution function from the equilibrium distribution function

$$\sigma_{\alpha\beta} = \left(1-\frac{1}{2\tau}\right) \sum_i \left(f_i - f_i^{eq} \right) c_{i\alpha} c_{i\beta} - \tfrac{1}{2}\left(1-\frac{1}{2\tau}\right)\left(F_\alpha u_\beta + F_\beta u_\alpha\right) \tag{4.6}$$

We can summarize the above by going through a lattice-Boltzmann time step, supposing that the force F_α is known: (1) Calculate density and velocity by Eqs. 4.4 & 4.5; (2) determine the equilibrium distribution (that involves ρ and u_α only); (3) determine Ω_i and S_i, the latter with Eq. 4.3; (4) perform the collision, Eq. 4.2; (5) stream.

Immersed boundary method
An important application of forcing fluid flow is the immersed boundary method (IBM). In this method, we attempt to achieve a certain flow velocity and a certain location in the flow domain by exerting a force on the fluid in the direct vicinity of that location. The "certain location" does not have to be a lattice node. This is the most attractive part of the IBM: it – in principle – removes the "squareness" of the flow domain and allows for curved boundaries without the need to approximate them as stair steps. IBM is an active area of research, not only in relation to the LB method but also in relation to other numerical schemes for fluid mechanics. I will be discussing here the way I have implemented the IBM in the LB method, which only is one of many ways. The discussion will involve two dimensions only; extension to 3D is not problematic.

Suppose we immerse a cylinder (a circle in 2D) in a fluid. The cylinder does not have to be fixed, it is given a known linear velocity $\mathbf{v_p}$ and a known angular velocity $\boldsymbol{\omega}_\mathbf{p}$ (note that in 2D the angular velocity has only one component so that – strictly speaking – we do not need to write it as a vector). The centre location of the cylinder we call $\mathbf{x_p}$. The cylinder is represented by points (*marker points*) on its surface, see Figure 4.1. The spacing between these points is an important choice to be made. As a rule of thumb, the spacing is taken as somewhat less than the lattice spacing Δ (e.g. 0.7Δ). We now focus on one of the marker

points that has location $\mathbf{x}_j(t)$ (j is the counter of marker points, since the cylinder is translating and rotating the location depends on time). The marker points is always surrounded by four lattice points (see Figure 4.1). On these lattice points the fluid velocity is known, let's call it $\mathbf{u}_i$ (i is the counter of lattice points; we use a single counter here for simplicity). We can interpolate this fluid velocity to location $\mathbf{x}_j(t)$. Given that we only involve four surrounding lattice points, this is a linear interpolation. This again is a choice; higher order interpolations are possible and would need to involve more lattice points. The interpolation process is written as

$$\mathbf{w}_j = \sum_i I(\mathbf{r}_{ij})\mathbf{u}_i \tag{4.7}$$

The sum is over the four surrounding lattice points, $I(\mathbf{r}_{ij})$ are interpolation coefficients that depend on the location $\mathbf{x}_j$ relative to the lattice points $\mathbf{x}_i$: $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$. Since we want to impose a no-slip condition at the surface of the cylinder, the fluid velocity at $\mathbf{x}_j$ should be equal to the surface velocity of the cylinder at $\mathbf{x}_j$ which is $\mathbf{v}_j = \mathbf{v}_p + \boldsymbol{\omega}_p \times (\mathbf{x}_j - \mathbf{x}_p)$. In general, the interpolated liquid velocity $\mathbf{w}_j$ at point $\mathbf{x}_j$ does not match $\mathbf{v}_j$. In order to force $\mathbf{w}_j$ to closely match $\mathbf{v}_j$, we exert a force on the fluid that acts against the velocity difference $\mathbf{w}_j - \mathbf{v}_j$. This is done in the form of a relaxation process:

$$\mathbf{F}_j = \alpha \mathbf{F}_j^{old} - \beta(\mathbf{w}_j - \mathbf{v}_j) \tag{4.8}$$

Relaxation means that we involve the force from the previous time step $\mathbf{F}_j^{old}$ in calculating the new force $\mathbf{F}_j$. The relaxation process involves two constants that I – in all honesty – needed to find by trial and error. In practice $\alpha = 0.95, \beta = 1.8$. One should now realize that the force $\mathbf{F}_j$ acts on a marker point, not on a lattice point. In a subsequent operation, $\mathbf{F}_j$ is distributed over the four lattice points that surround $\mathbf{x}_j$. For consistency, this force distribution step uses the same coefficients as the interpolation step

$$\mathbf{F}_i = I(\mathbf{r}_{ij})\mathbf{F}_j \tag{4.9}$$

with $\mathbf{F}_i$ the force on lattice node i. This then is the force that we can use in the lattice-Boltzmann scheme discussed above.

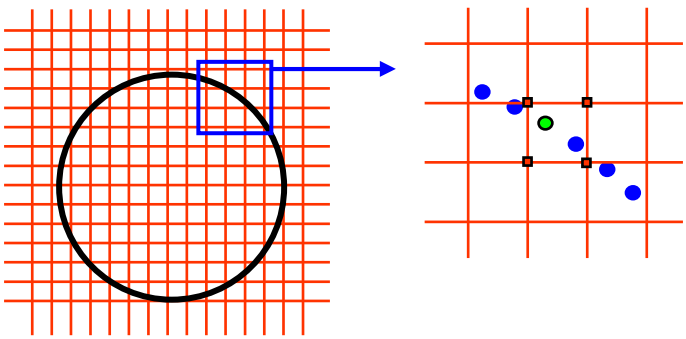

Figure 4.1 Immersed boundary method: cylinder on a square grid. Green and blue dots: marker points. Interpolation to the green dot from the four surrounding (red) lattice nodes.

It is useful to note that this procedure, in addition to imposing no-slip on immersed surfaces, also provides us with the force exerted on the immersed surface. The immersed boundary force we calculate according to the procedure above is the force exerted by the surface on the fluid. Since action equals minus reaction, the force on the surface is the opposite of the immersed boundary force. Adding up and inverting the immersed boundary forces for all marker points thus gives us the total force on the immersed surface; in the example of the cylinder this would be the hydrodynamic force exerted on the cylinder.

4.3 The multiple relaxation time (MRT) collision operator

For various reasons, the BGK collision operator, which is based on a single relaxation time, has limited stability and accuracy. This is specifically apparent for large velocity magnitudes (Mach numbers not being very small) and/or low viscosities, i.e. relaxation times only slightly larger than 0.5. One of the consequences is that it is hard to achieve high Reynolds numbers without building very large grids of lattice nodes. One issue that limits the accuracy of LB-BGK simulations at relatively high fluid speeds is the term containing $u_\alpha u_\beta u_\gamma = O\left(u^3\right)$ that appeared in the Chapman-Enskog analysis (see LN02, Eq. 2.28).

In analysing the BGK collision operator, it has been recognized that it is a rather crude idea to apply the same relaxation time to all distribution functions. Generalizations of the BGK operator are therefore based on the idea of involving more than one relaxation time. To make this physically meaningful, relaxation times are assigned to *velocity moments* of the distribution function, instead of to distribution functions themselves. Examples of velocity moments of the distribution function are the density as the zeroth moment and momentum as the first moment: $\rho = \sum_i f_i$ and $\rho\mathbf{u} = \sum_i \mathbf{c}_\mathbf{i} f_i$ respectively.

Expressing velocity moments in terms of distribution functions is a linear operation and can be cast in a matrix-vector form:

$$\mathbf{m} = \mathbf{M} \cdot \boldsymbol{f} \tag{4.10}$$

with $\boldsymbol{f}$ the vector containing distribution functions f_i ($\boldsymbol{f}$ has nine components in case of a D2Q9 lattice), $\mathbf{m}$ the vector containing the moments (has the same dimension as $\boldsymbol{f}$) and $\mathbf{M}$ a square matrix with constant coefficients.

The book “The Lattice Boltzmann Method – Principles and Practice” (referred to in LN01) gives an excellent explanation, that we closely follow here, of how the BGK operator is generalized to a multiple relaxation time (MRT) operator. We introduce a new parameter $\omega = 1/\tau$ with τ the BGK relaxation time. The lattice-Boltzmann BGK equation reads

$$f_i\left(\mathbf{x}+\mathbf{c}_\mathbf{i}, t+1\right) - f_i\left(\mathbf{x}, t\right) = -\omega\left[f_i\left(\mathbf{x}, t\right) - f_i^{eq}\left(\mathbf{x}, t\right)\right] \tag{4.11}$$

In ($\boldsymbol{f}$) vector form it thus reads

$$\boldsymbol{f}\left(\mathbf{x}+\mathbf{c}_\mathbf{i}, t+1\right) - \boldsymbol{f}\left(\mathbf{x}, t\right) = -\omega\left[\boldsymbol{f}\left(\mathbf{x}, t\right) - \boldsymbol{f}^{eq}\left(\mathbf{x}, t\right)\right]$$

multiply the right-hand side by the identity matrix $\mathbf{I} = \mathbf{M}^{-1}\mathbf{M}$

$$\boldsymbol{f}\left(\mathbf{x}+\mathbf{c}_\mathbf{i},t+1\right)-\boldsymbol{f}\left(\mathbf{x},t\right)=-\mathbf{M}^{-1}\mathbf{M}\omega\left[\boldsymbol{f}\left(\mathbf{x},t\right)-\boldsymbol{f}^{eq}\left(\mathbf{x},t\right)\right]$$

realize ω is a constant

$$\boldsymbol{f}\left(\mathbf{x}+\mathbf{c}_\mathbf{i},t+1\right)-\boldsymbol{f}\left(\mathbf{x},t\right)=-\mathbf{M}^{-1}\omega\left[\mathbf{M}\boldsymbol{f}\left(\mathbf{x},t\right)-\mathbf{M}\boldsymbol{f}^{eq}\left(\mathbf{x},t\right)\right]$$

with Eq. 4.10:

$$\boldsymbol{f}\left(\mathbf{x}+\mathbf{c}_\mathbf{i},t+1\right)-\boldsymbol{f}\left(\mathbf{x},t\right)=-\mathbf{M}^{-1}\omega\left[\mathbf{m}\left(\mathbf{x},t\right)-\mathbf{m}^{eq}\left(\mathbf{x},t\right)\right]$$

multiply yet again by the identity matrix

$$\boldsymbol{f}\left(\mathbf{x}+\mathbf{c}_\mathbf{i},t+1\right)-\boldsymbol{f}\left(\mathbf{x},t\right)=-\mathbf{M}^{-1}\omega\mathbf{I}\left[\mathbf{m}\left(\mathbf{x},t\right)-\mathbf{m}^{eq}\left(\mathbf{x},t\right)\right]$$

define the diagonal matrix $\mathbf{S} = \omega\mathbf{I}$

$$\boldsymbol{f}\left(\mathbf{x}+\mathbf{c}_\mathbf{i},t+1\right)-\boldsymbol{f}\left(\mathbf{x},t\right)=-\mathbf{M}^{-1}\mathbf{S}\left[\mathbf{m}\left(\mathbf{x},t\right)-\mathbf{m}^{eq}\left(\mathbf{x},t\right)\right]$$

The generalization that we make now is to replace $\mathbf{S} = \omega\mathbf{I}$ by a diagonal matrix $\mathbf{S}$ that has relaxation rates (i.e. inverse relaxation times) per moment on the diagonal (for a nine-velocity set)

$$\mathbf{S}=\begin{bmatrix}\omega_0 & 0 & \cdots & 0\\ 0 & \omega_1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \omega_8\end{bmatrix} \tag{4.12}$$

We now need to figure out which velocity moments to select and how to determine their respective relaxation rates.

For selecting the moments we go by the so-called Gram-Schmidt procedure that – for the D2Q9 velocity set as introduced in LN02 – results in the following transformation matrix

$$\mathbf{M}=\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2\\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1\\ 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1\\ 0 & -2 & 2 & 0 & 0 & 1 & -1 & -1 & 1\\ 0 & 0 & 0 & 1 & -1 & 1 & 1 & -1 & -1\\ 0 & 0 & 0 & -2 & 2 & 1 & -1 & 1 & -1\\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1\end{bmatrix} \tag{4.13}$$

The Gram-Schmidt equilibrium moments are

$$\begin{gathered}0:\rho^{eq}=\rho \quad 1:e^{eq}=\rho-3\rho\left(u_x^2+u_y^2\right) \quad 2:\varepsilon^{eq}=9\rho u_x^2-3\rho\left(u_x^2+u_y^2\right)+\rho\\ 3:j_x^{eq}=\rho u_x \quad 4:q_x^{eq}=3\rho u_x^3-\rho u_x \quad 5:j_y^{eq}=\rho u_y \quad 6:q_y^{eq}=3\rho u_y^3-\rho u_y\\ 7:p_{xx}^{eq}=\rho\left(u_x^2-u_y^2\right) \quad 8:p_{xy}^{eq}=\rho u_x u_y\end{gathered} \tag{4.14}$$

The relaxation rates are contained in the diagonal matrix **S**:

$$\mathbf{S} = \mathrm{diag}\left(0, \omega_e, \omega_\varepsilon, 0, \omega_q, 0, \omega_q, \omega_\nu, \omega_\nu\right) \tag{4.15}$$

The conserved velocity moments (density and momentum) have zero relaxation rates. The rates ω_e and ω_ν determine bulk and shear viscosity

$$\nu = c_s^2\left(\frac{1}{\omega_\nu} - \tfrac{1}{2}\right) \quad \nu_B = c_s^2\left(\frac{1}{\omega_e} - \tfrac{1}{2}\right) - \frac{\nu}{3} \tag{4.16}$$

(note that – different from BGK – we now have the means to independently set bulk and shear viscosity). The rates ω_ε and ω_q are "free" parameters. For the D2Q9 velocity set ω_ε=1 and $\omega_q = 1$ give good results.

An algorithm performing LB simulations with an MRT collision operation would go as follows: Start from a pre-collision distribution $f_i(\mathbf{x},t)$

(1) determine density and velocity in the usual way $\rho = \sum_i f_i$ and $\rho u_\alpha = \sum_i f_i c_\alpha$ (assuming you do not have body forces);

(2) from density and velocity determine $\mathbf{m}^{eq}(\mathbf{x},t)$ (Eq. 4.14);

(3) from $\boldsymbol{f}(\mathbf{x},t)$ determine $\mathbf{m}(\mathbf{x},t)$ (Eq. 4.10);

(4) determine the post collision distribution $\boldsymbol{f}^*(\mathbf{x},t) = -\mathbf{M}^{-1}\mathbf{S}\left[\mathbf{m}(\mathbf{x},t) - \mathbf{m}^{eq}(\mathbf{x},t)\right]$ (this requires determining $\mathbf{M}^{-1}$ which you only need to do once & which Matlab can do for you)

(5) stream: $f_i^*(\mathbf{x}+\mathbf{c_i}, t+1) = f_i(\mathbf{x},t)$.