OpenLB technical report: BGK dynamics in which average density is time-independent

Jonas Latt

April 2008

1 Density variations in lattice Boltzmann

In an ideal gas, pressure is proportional to the particle density, with the speed of sound c_s as proportionality factor:

$$p = c_s^2 \rho. \tag{1}$$

The density (or pressure) can be split into a thermodynamic component ρ_0, which is space-independent but may depend on time, and a hydrodynamic component ρ_h, whose contribution scales as the square of the Mach number Ma:

$$\rho(x,t) = \rho_0(t) + \mathrm{Ma}^2 \rho_h. \tag{2}$$

Lattice Boltzmann models are locally mass conserving. Therefore, ρ_0 is time-independent if the boundary condition is globally mass-conserving (for example, if all boundary nodes use full-way bounce-back). But, when Dirichlet velocity boundaries are implemented (and with some boundary conditions, even with no-slip boundaries), global mass-conservation is not guaranteed. Of course, a boundary condition for an incompressible fluid must conserve mass asymptotically:

$$\int_{\partial\Omega} \rho u \cdot n d\sigma = 0, \tag{3}$$

where $\partial\Omega$ is the domain boundary, n the boundary normal and $d\sigma$ a line element along the boundary. In an implementation however, this global conservation law may be subject to numerical approximation. A particularly striking example in which mass is not conserved in a lattice Boltzmann simulation is the cavity flow. Here, the velocity profile is discontinuous in the two upper corners, which results in a visible non-zero mass balance.

Conceptually speaking, time-variations of ρ_0 are not an issue in incompressible fluids, because only the gradient of the pressure enters the Navier-Stokes equations, and not the value of p itself. Such variations are however problematic in actual simulation, because lattice Boltzmann does for example not support negative density values. Furthermore, the accuracy of simulations generally decreases when ρ deviates too strongly from 1. Therefore, global density variations due to boundary conditions should be avoided.

2 The `ConstRhoBGKdynamics` in OpenLB

A way to circumvent these technical difficulties is to reset ρ_0 to 1 after each iteration step. Which means:

1. Compute the average density $\bar{\rho}$ over the lattice.
2. Subtract $\bar{\rho} - 1$ from every lattice site.

The following line of code does the required operation in OpenLB:

```
lattice.stripeOffDensityOffset(lattice.computeAverageDensity()-1)
```

This works, but is not efficient. If you do this, you need to traverse memory three times per iteration step instead of only once (see TR1): once for collision-propagation, once to compute the average density and once to stripe off the density offset. Things would be much more efficient if this density correction were handled within the collision step. That's exactly what happens when you use `ConstRhoBGKdynamics` instead of `BGKdynamics`, where the $\bar{\rho} - 1$ correction is integrated into collision. To know the value of $\bar{\rho}$, the internal statistics of the lattice is accessed. That means that the value of $\bar{\rho}$ at the previous iteration is used, not the current one. Therefore, ρ_0 is not *exactly* reset to 1, but almost. This certainly prevents you from reaching a numerically problematic regime for ρ, and it is numerically almost as efficient as the original BGK model.

3 Discussion

The `ConstRhoBGKdynamics` model can save a simulation from numerical instability, but bears the danger of being overused. An important assumption of the model is that the density balance introduced by the boundary is essentially constant in space and does not scale with the Mach number. This assumption is violated if a boundary injects density at a high rate, or if it produces pressure waves. In that case, the boundary condition is invalid and needs to be fixed: `ConstRhoBGKdynamics` doesn't do any good here.