Lattice-Boltzmann Method Used to Model Elastic Objects in Fluid

Iveta Jančigová*
Department of Information Networks
Faculty of Management Science and Informatics
UNIZA, Slovakia
iveta.jancigova@fri.uniza.sk

***Abstract*—The lattice Boltzmann method (LBM) has been widely used to model various fluid systems. The treatment of boundary conditions is still an ongoing topic, especially for situations when a lot of moving objects is involved. In this paper we briefly explain the method itself and then focus on how to incorporate direct forcing into it, in order to ensure the no-slip boundary condition on the surfaces of immersed bodies. This approach to modeling elastic objects in fluid avoids solving the more computationally expensive Navier-Stokes equations and also the bounce-back condition typical for LBM. We then discuss the possible implementations of the direct forcing and planed work (validation of results).**

***Keywords*—blood flow, elastic object, lattice Boltzmann method, immersed boundary method, direct forcing**

I. Introduction

THE particle-fluid interaction problem arises in many applications in chemistry, geology, environmental engineering as well as in biology. The motivation for our efforts are the microfluidic devices as proposed in [5]. These are used for efficient and selective separation of the circulating tumour cells from peripheral blood samples of patients with cancer. Our ultimate goal is to optimize the design of such devices, but for that we need a good model of blood plasma (fluid) and the objects traveling in it (e.g. red blood cells, circulating tumor cells).

Conventional methods for simulation of particulate flows with a large number of immersed objects, such as Navier-Stokes equations solved using finite element method, are not a very good choice in this situation because of the boundary conditions resulting from all the moving elastic objects. The lattice Boltzman method (LBM), first introduced in early 90s [3], overcomes this limitation because it uses a fixed grid to represent the flow field. This grid does not need to be re-meshed during computation and therefore is much more efficient. The interaction between fluid and elastic objects is done using the immersed boundary (IB) method. In order to properly model the no-slip boundary condition, we implemented direct forcing as suggested in [2].

II. Lattice-Boltzmann Method

The LBM uses a regular grid and represents the fluid domain by a set of lattice nodes. In our 3-dimensional case this means a cubic lattice with 19 discrete velocity directions known as D3Q19. The fluid itself is modeled as a group of fictitious fluid particles that are only allowed to either stay where they are or move to the neighboring nodes. The state of the fluid at a given node $\mathbf{x}$ at time t is described by distribution functions $f_a(\mathbf{x}, t)$ and the governing equation consists of streaming and collision (i.e. relaxation towards local equilibrium) [7]:

$$f_a(\mathbf{x}+\mathbf{e}_a, t+1) = f_a(\mathbf{x}, t) - \frac{f_a(\mathbf{x}, t) - f_a^{eq}(\mathbf{x}, t)}{\tau} \quad (1)$$

$$f_a^{eq}(x) = w_a \rho(\mathbf{x}, t)\left[1 + 3\frac{\mathbf{e}_a \cdot \mathbf{u}}{c^2} + \frac{9}{2}\frac{(\mathbf{e}_a \cdot \mathbf{u})^2}{c^4} - \frac{3}{2}\frac{\mathbf{u}^2}{c^2}\right] \quad (2)$$

where a represents the directions to neighboring nodes, $\mathbf{e}_a$ are the velocity vectors pointing to the adjacent nodes, τ is the relaxation scale related to viscosity through $\nu = \frac{1}{6}(\tau - 1)$, ρ is the fluid density, w_a are the lattice weights ($w_0 = \frac{1}{3}, w_{1..6} = \frac{1}{18}, w_{7..18} = \frac{1}{36}$), $\mathbf{u}$ is the macroscopic velocity at the given node and c is the lattice speed of sound $c = \frac{1}{\sqrt{3}}$.

This is an Eulerian specification of the flow field and at each node of the lattice, the density and velocity can be written in terms of the distribution functions f_a:

$$\rho(\mathbf{x}, t) = \sum_a f_a(\mathbf{x}, t) \quad (3)$$

$$\mathbf{u} = \frac{1}{\rho(\mathbf{x}, t)} \sum_a f_a(\mathbf{x}, t)\mathbf{e}_a \quad (4)$$

Using Chapman-Enskog expansion, one can recover the macroscopic continuity and momentum (Navier-Stokes) equations form (1)-(4) [6].

III. Immersed Boundary Method

The greatest challenge of using the LBM for fluid-particle interaction problems is to properly incorporate the boundary conditions. That is the role of the immersed boundary method which treats the boundary points on the surface of the immersed object as Lagrangian particles. In our case, these are obtained from the triangulation of the surface of the objects, they move as the object moves and in general they

*This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0441-11.

do not correspond to the lattice nodes.

For us, the objects in question are red blood cells, modeled as a network of strings on their surface [1]. Several conditions are applied (on stretching, bending, conservation of area both globally and locally, conservation of volume), to make sure that the cell's properties correspond to biology. Even though in [1] very good results are obtained when the objects are propagated forward using drag force (using a software package ESPResSo [4] adapted from molecule simulations to simulations of closed objects), the consequence of this approach is a non-zero difference of velocities of the fluid and particle on the surface.

In order to improve the model, we decided to implement the no-slip boundary condition in the same setting. This can be done using the bounce-back condition [8], which is computationally a very expensive approach or alternatively using the direct forcing method [2].

IV. Direct Forcing

The direct forcing method is based on the Navier-Stokes equations which could also be used to govern the fluid-particle system:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)=-\nabla\mathbf{p}+\nu\nabla^2\mathbf{u}+\mathbf{f} \tag{5}$$

In these equations, $\mathbf{p}$ is the pressure of the fluid and $\mathbf{f}$ is the force density. Equation (5) is also valid at the Langrangian boundary points and may be written as follows:

$$\mathbf{f}=\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)-\nu\nabla^2\mathbf{u}+\nabla\mathbf{p} \tag{6}$$

This equation is then discretized (Einstein notation for subscripts and derivatives is used)

$$f_i^{(n+1)}=\rho_i\left(\frac{u_i^{(n+1)}-u_i^{(n)}}{\Delta t}+u_j^{(n)}u_{j,i}^{(n)}\right)-\nu u_{i,jj}^{(n)}+p_{,i}^{(n)}$$

and it is assumed that the velocity and pressure are known at the time $t=t_n$.

In order to impose the no-slip boundary condition - that is to make sure that at time $t=t_{n+1}=t_n+\Delta t$, the velocity on the immersed Lagrangian boundary points is the same as the velocity of the fluid at this point - we force it by adjusting this discretized density force to reflect this:

$$f_i^{(n+1)}=\rho_i\left(\frac{U_i^{P(n+1)}-u_i^{(n)}}{\Delta t}+u_j^{(n)}u_{j,i}^{(n)}\right)-\nu u_{i,jj}^{(n)}+p_{,i}^{(n)}$$

Here $U_i^{P(n+1)}$ is the velocity of the particle on the surface of the immersed object and when this force is applied (to the particle and also to the fluid at the same point but in the opposite direction), in the next step their velocities (without other influence) would be equal.

The question arises, how exactly to implement this, since the lattice nodes and the surface nodes do not coincide. One approach is to perform the computations at the surface. For that it is necessary for each surface particle to interpolate the fluid velocity from the neighboring 8 lattice nodes and then distribute the computed force back to them (This can be done using the Dirac delta functions). We chose the other approach, namely to distribute the surface particle velocity to the neighboring lattice nodes first. There we compute the interaction force and then interpolate it back to the surface particle. The interpolation is done using the partial volumes as weights. For all the space derivatives we used second order finite differences.

The simulation package ESPResSo does not directly work with pressure. Since we needed pressure for the direct forcing, two approaches were tried - calculation of pressure as an average of the normal components of the stress tensor and and computing the pressure locally from fluid density as $p=\rho c^2$, where this c is the lattice speed of sound that also appears in the LB equation (2).

V. Future Work

More work here is needed because the two approaches of pressure computation should in theory give similar results and as of now they do not. In order to check the stress tensor computation, we plan to perform three simulation experiments:

A. *Hydrostatic pressure*

We will look at an empty channel with no flow, measure the hydrostatic pressure and compare it with theoretical formula at various depths.

B. *Couette flow*

We will model two plates, one of which is stationary and one is being moved resulting in laminar flow with constant shear stress across the domain. We hope to implement this by not actually moving the wall itself, but rather the layer of fluid right next to it. We will do this by designating the two plates to be opposite side walls and having periodic boundary conditions on top, bottom, front and back walls.

C. *Moving sphere/ellipsoid slowing down in stationary fluid*

This is a problem with known analytic solution, which has one degree of freedom - the mass. In our model, the mass is distributed equally among the particles on the surface and this experiment can be used for its calibration. However, there is still an open question, whether it is reasonable approximation to have all the mass on the surface instead of throughout the whole volume.

Once we get reasonable reassurance that the pressure is working as it should, we will perform comparison tests with the object propagation done using drag force. We would like our approach to be more efficient, since that together with more accurate physical representation would warrant further use of this method in our long-term efforts.

VI. Conclusion

The lattice Boltzmann method with direct forcing is a reasonable alternative for modeling no-slip boundary conditions to the bounce-back approach. While still more work is needed, the basic idea looks promising and if it turns out to be efficient enough, it could replace the movement based on drag force in ESPResSo simulations of closed elastic objects in fluid.

Acknowledgment

The author would like to thank Dr. Ivan Cimrák and doc. RNDr. Katarína Bachratá, PhD for valuable guidance and support.

References

[1] I. Cimrák, M. Gusenbauer, T. Schrefl, *Modeling and simulation processes in microfluidic devices for biomedical applications*, Computers and Mathematics with Applications 64(3), 2012.

[2] Z.-G. Feng, E.E. Michaelides, *Proteus: A direct forcing method in the simulations of particulate flows*, J. Comput. Phys. 202(1), 2005.

[3] A.J.C. Ladd, *Numerical simulations of particulate suspensions via discretized Boltzmann equation*, Part I, II, J. Fluid Mech. 271, 1994.

[4] H.-J. Limbach, et al., *ESPResSo An Extensible Simulation Package for Research on Soft Matter Systems*, Comput. Phys. Commun. 174(9), 2006.

[5] S. Nagrath, et al., *Isolation of rare circulating tumour cells in cancer patients by microchip technology*, Nature 450, 2007.

[6] S. Succi, *The lattice Boltzmann equation*, Oxford University Press, 2001.

[7] M.C. Sukop, D.T. Thorne, *Lattice Boltzmann modeling: An introduction for geoscientists and engineers*, Springer, 2nd ed., 2007.

[8] X. Yin, J. Zhang, *An improved bounce-back scheme for complex boundary conditions in lattice Boltzmann method*, J. Comput. Phys. 231(11), 2012.