Journal of Computational Science 9 (2015) 45–50

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Parallel performance of an IB-LBM suspension simulation framework

Lampros Mountrakis[a,*], Eric Lorenz[a,b], Orestis Malaspinas[c,d], Saad Alowayyed[a,e], Bastien Chopard[c], Alfons G. Hoekstra[a,f]

[a] Computational Science, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands
[b] Electric Ant Lab B.V., Panamalaan 4K, 1019 AZ, Amsterdam, The Netherlands
[c] Computer Science Department, University of Geneva, 7 route de Drize, 1227 Carouge, Switzerland
[d] Institut d'Alembert, UMR CNRS 7190, Université Pierre et Marie Curie – Paris 6, 4, place Jussieu, case 162, F-75252 Paris cedex 5, France
[e] King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia
[f] National Research University ITMO, Saint-Petersburg, Russian Federation

ARTICLE INFO

Article history:
Available online 17 April 2015

Keywords:
LBM
IBM
Parallel
MPI
Suspension
Blood
Palabos

ABSTRACT

We present performance results from `ficsion`, a general purpose parallel suspension solver, employing the Immersed-Boundary lattice-Boltzmann method (IB-LBM). `ficsion` is built on top of the open-source LBM framework `Palabos`, making use of its data structures and their inherent parallelism. We describe in brief the implementation and present weak and strong scaling results for simulations of dense red blood cell suspensions. Despite its complexity the simulations demonstrate a fairly good, close to linear scaling, both in the weak and strong scaling scenarios.

1. Introduction

Blood is a substance where the microstructure plays an important role in understanding the rheology and transport properties of this dense suspension. Approximately 5 million deformable red blood cells (RBCs) per cubic millimeter account for 40–45% of the total blood volume. They bring out many biologically interesting phenomena like the *Fåhræus* and *Fåhræus-Lindqvist* effects [1,2], the margination of platelets [3] and the non-Newtonian nature of blood [4]. Blood flow models can give insights to studies of cell diseases, such as malaria or sickle-cell anemia [5,6] and also in cardiovascular diseases like the formation of atherosclerotic plaques or thrombosis in aneurysms [7–10].

Simulations of dense suspensions, like blood, demand considerable computational resources. Software, in terms of algorithms, data structures and parallelism, is becoming more and more crucial in extracting knowledge from such systems. Implementing basic algorithms is relatively straightforward, yet complexity steeply increases by incorporating parallelism, elaborate boundary conditions, or other advanced elements, such as thermal and multiphase flows, moving objects, and suspended particles. A number of lattice-Boltzmann solvers already have several of the aforementioned capabilities implemented and tested and are released under an open-source license. Some examples of such established frameworks are, e.g. `Palabos` [11], `LB3D` [12], `Sailfish` [13], `HemeLB` [14], `LUDWIG` [15] and `Musubi` [16].

In this work we present the parallel performance of `ficsion`, a general purpose parallel IB-LBM solver with a focus on suspensions of deformable particles, like blood. `ficsion` is build on top of the open-source `C++` framework `Palabos` [11], making use of its data structures, parallelism and well-tested modules. The development of a fully parallelized suspension code implemented on top of a third-party framework is a challenging task, especially when the developer has no direct control over parallelization, and the existing parallel data structures have to be employed in a creative way. In this paper we briefly describe the implementation behind `ficsion` and present weak and strong scaling results for its application to the simulation of fully resolved blood flow.

2. Methods

Our approach is based on the immersed boundary-lattice Boltzmann method (IB-LBM), a combination frequently used in modeling blood suspensions [17–19]. Suspensions of deformable cells are the

* Corresponding author.
E-mail addresses: L.Mountrakis@uva.nl (L. Mountrakis), E.Lorenz@electricant.com (E. Lorenz), Orestis.Malaspinas@unige.ch (O. Malaspinas), S.A.Alowayyed@uva.nl (S. Alowayyed), Bastien.Chopard@unige.ch (B. Chopard), A.G.Hoekstra@uva.nl (A.G. Hoekstra).

http://dx.doi.org/10.1016/j.jocs.2015.04.006
1877-7503/

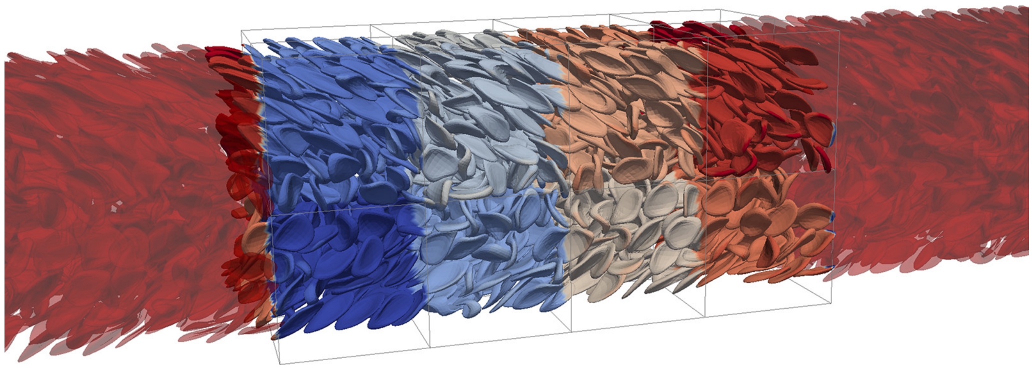

Fig. 1. Snapshot of channel flow between two parallel plates, placed on the top and bottom of the channel. The domain is periodic in the other dimensions. Size of the domain is $128 \times 64 \times 64\,\mu m^2$ with ≈1530 RBCs (hematocrit $H \approx 30\,\%$). The simulation was performed in 16 cores and color denotes the MPI rank. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

focus of `ficsion`, yet methods for hard-objects, like the Noble-Torczynski [20] or Ladd's method [21], could be incorporated and parallelism would be retained.

RBCs are represented as a mesh of Langrangian surface points, interacting via a spectrin link model. LBM is used for the fluid phase and is coupled to the suspended RBCs with the immersed boundary method. Fig. 1 depicts a snapshot from a representative simulation of blood flow between two parallel plates with `ficsion` using $N_p = 16$ processors. Periodic boundary conditions were used for this simulation, with a body force acting as a pressure gradient.

2.1. Lattice Boltzmann method

The lattice Boltzmann methods (LBMs) [22] are a class of mesoscopic particle-based approaches simulating fluid-flow. The local kinetic scheme of LBM allows for an intrinsic parallelization, rendering it a good candidate for parallel computing [23,24].

The main quantity of a lattice Boltzmann model is the population density $f_i(\mathbf{x}, t)$, which corresponds to the discretized probability distribution of finding fluid particles at site $\mathbf{x}$ and time t, moving with a discrete velocity $\mathbf{c}_i$. The general form of a LBM is:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \Omega_i(f_i(\mathbf{x}, t)) \quad (1)$$

where $\Omega_i(f_i(\mathbf{x}, t))$ is the collision operator, which re-shuffles densities f_i according to the kinetic theory of gases and Δt is the time step.

A wide range of models for a variety of applications has been developed starting from the general description. A simple and widely used form of $\Omega_i(f_i(\mathbf{x}, t))$ is the linearized single relaxation time collision operator, or LBGK model. In LBGK $\Omega_i(f_i(\mathbf{x}, t)) = -\frac{1}{\tau}(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t))$, where τ is the relaxation parameter and $f_i^{eq}(\mathbf{x}, t)$ the equilibrium population corresponding to an expansion of the Maxwell–Boltzmann distribution for small Mach numbers. The zeroth and first moment of the population densities recover the fluid density ρ and velocity $\mathbf{u}$ according to $\rho = \sum_i f_i$ and $\rho\mathbf{u} = \sum_i f_i \mathbf{c}_i$. Kinematic viscosity ν for the LBGK collision operator is given by $\nu = (\tau - \frac{1}{2})c_s^2 \Delta t$ in which $c_s = \frac{1}{\sqrt{3}}\Delta x/\Delta t$ is the lattice speed of sound.

2.2. Immersed boundary method

The immersed boundary method (IBM) [25] is a pure coupling method used in fluid structure interaction. One of the main advantages of IBM, is that the fluid and the immersed structure do not need to conform. This alleviates the need for remeshing and renders complex configurations like dense suspensions easier to handle. With IBM the *cell*[1] follows the Lagrangian description.

The basic concept of IBM is the no-slip condition at the interface of the membrane and the fluid. This is realized as the Langrangian *surface points* (or *surface particles*) which constitute the membrane mesh, exert a force to the fluid, while they are advected by interpolating the fluid velocity.

The surface particle $\mathbf{x}_i(t)$ spreads a force $\mathbf{F}_i(t)$ to the closest Eulerian points $\mathbf{X}$ of the fluid according to

$$\mathbf{f}(\mathbf{X}, t) = \sum_i \mathbf{F}_i(t)\delta(\mathbf{X} - \mathbf{x}_i(t)) \quad (2)$$

where $\delta(\mathbf{X} - \mathbf{x}_i(t))$ is a discrete Dirac delta function. Following this step, the position of the particle is updated according to the Eulerian scheme

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{u}_i(t + \Delta t)\Delta t \quad (3)$$

where

$$\mathbf{u}_i(t + \Delta t) = \sum_{\mathbf{X}} \mathbf{u}(\mathbf{X}, t + \Delta t)\delta(\mathbf{X} - \mathbf{x}_i(t)), \quad (4)$$

and uses the same discrete Dirac delta function $\delta(\mathbf{X} - \mathbf{x}_i(t))$ as in Eq. (2).

$\delta(\mathbf{r})$ is constructed by multiplying 1D interpolation kernel functions ϕ, as $\delta(\mathbf{r}) = \phi(x)\phi(y)\phi(z)$. It is possible to find many discretized delta functions with varying interpolation ranges, yet a number of restrictions should be fulfilled [25]. In the present work we used the kernel $\phi_2(r)$:

$$\phi_2(r) = \begin{cases} 1 - |r| & |r| \le 1, \\ 0 & |r| \ge 1 \end{cases} \quad (5)$$

for its simplicity and compact support. The subscript of ϕ describes the support width of the kernel. More kernels are available and we have performed some benchmarks in 2D [19].

2.3. Membrane model of a single RBC

Blood consists of a vast number of red blood cells, biconcave disk-shaped membranes with a diameter of 6–8 μm and a thickness of approximately 2–2.5 μm. In the current work we employ the spectrin-link model to represent an RBC [26–29], in which a systematic coarse-graining procedure has been developed [27]

[1] The term *particle* is typically used in the literature of hard objects for what we here define as a *cell*. As *cell* we define the network of *surface particles* (or simply *particles*) that constitute the surface of the object and interact via a constitutive model.