Max Planck Society

Max Planck Institute for Intelligent Systems

Active Matter Group

Simulation of Q2D flow with LAMMPS

Author:
Rumen N. Georgiev

Supervisor:
William E. Uspal

October 26, 2016

I. Introduction

The self-assembly of micron-sized particles has drawn considerable attention over the past decade [1]. Their ability of forming phononic crystal has been a topic of debate and study [2]. Advances in photolithographic techniques have lead to the development of the stop-flow lithography method, which enables the fabrication of polyethylene glycol particles with a wide variety of shapes in microfluidic devices [3–5].

Recent work by Uspal, Eral and Doyle has focused on the theoretical aspect of these particles' self-organization. These researchers investigated the hydrodynamic interactions and collective motion of clusters of disks [6][7] and asymmetrical particles composed of disks held together by rigid shafts [8][7]. These particles, confined between the vertical walls of a microchannel, are set in motion when a pressure drop is applied across the channel. The thin lubrication layers separating the faces of the particles from the vertical walls increase the drag and the particles lag the surrounding fluid, thus introducing a flow disturbance field. It is this disturbance which gives rise to hydrodynamic interactions between the particles and causes their self-organisation. The system being 'approximately 2D' is dubbed quasi two-dimensional or q2D.

In their work, Uspal and Doyle obtain the fluid velocity field around a particle by numerically integrating the equations of motion. An alternative is to 'build' a system which contains a fluid and the particles of interest and observe its time evolution – in other words, one can also simulate the behaviour of the system.

Classical simulation techniques such as molecular dynamics, however, are not fitting for such a system, where there is a separation of both length and time scales – one needs to incorporate both the mesoscopic colloid, as well as the microscopic fluid particles. In such cases *coarse-grained* methods come into play, managing to reconcile this disparity. In coarse-grained simulations the detailed properties of one component, usually the fluid, are neglected and it is modelled in such a way that it still recovers the relevant physical phenomena while maintaining a low computational price. In our case this phenomenon is the fluid-mediated interaction between particles.

Two popular coarse-grained methods which recover hydrodynamics are Stochastical Rotational Dynamics (SRD)[9] and the Lattice Boltzmann Method (LBM)[10]. Both schemes reduce the degrees of freedom the fluid has by either confining groups of particles to a grid, like in LBM, or by significantly reducing the number of fluid particles present (SRD). Their applicability to q2D systems has not been compared, at least to our knowledge, and it is instructive to determine which of the two is the better-fitting tool when studying hydrodynamic interactions in q2D.

Since we are interested in both LBM and SRD we turn our attention to multi-purpose simulation packages, which support both. One such is the Large-scale Atomic/Molecular Massively Parallel Simulator, commonly known as LAMMPS [11][12][13]. What makes LAMMPS particularly convenient is its effective parallelization of both algorithms.

Having said this, we can outline the main goal of our work: to implement an asymmetric dumbbell particle in a coarse-grained fluid, track its motion and try to verify the experimental observations reported in [7]. To that end we use the SRD and LBM algorithms as implemented in the multi-purpose simulator LAMMPS.

This rest of the report is structured as follows:

1. Chapter II. provides the theoretical background for understanding fluid mechanics. Starting from the divergence theorem, mass and momentum conservation, we derive the Navier-Stokes equations for an incompressible fluid. Since we are interested in the flow in a microfluidic channel, we also derive the plane Poiseuille profile. This chapter draws inspiration from the works of H. Bruus and his course in theoretical microfluidics [14].

2. In Chapter III. we introduce the two methods we use to simulate Q2D flow in LAMMPS – Lattice Boltzmann and Stochastic Rotational Dynamics. We describe both algorithms in detail and, by introducing important quantities like the Reynolds and Schmidt numbers, we justify our choice

of simulation parameters. Our concise review of the two methods is based on reviews by Padding and Louis [15] (SRD) Gompper *et al.* [16] (SRD) and Chen and Doolen [17](LBM).

3. Our simulation results are presented in Chapter IV. The fluid flow field around a dumbbell is obtained with the Lattice Boltzmann package. We manage to observe rotation of the colloid but due to computational limitation we cannot draw quantitative conclusions. We try to couple a non-spherical colloid to the SRD fluid but this attempt is not successful.

4. In Chapter V. we discuss the results we obtain and identify the drawbacks of both packages, while proposing possible remedies.

5. We conclude our report with Chapter VI.

II. Theory

A. Fluid Mechanics

Full description of any fluid flow requires the introduction of the Navier-Stokes equations (NSE). They are a set of non-linear partial differential equations used to obtain the velocity field under some initial and boundary conditions. For some simple cases (flow of pure incompressible fluid in a slit, flow around a sphere in three-fold periodic boundary conditions, etc.) one is able to obtain an analytical solution. This task, however, is overwhelmingly difficult for the system we are interested in and in such cases one is forced to resort to numerical calculations. Simulating a simple system and comparing the computational result to the analytical one is a common way of verifying a proposed algorithm. A close match between the two results serves as basis for further augmentations of the system.

In this section we obtain the NSE by first deriving the continuity equation, *i.e.*, how system properties are conserved. Then we use this equation to enforce mass and momentum conservation of our system. Finally, introduce the idea of the stress tensor and with its help derive the final form of the NSE.

1. Continuity Equation

We will start by defining a volume Q enclosed by a surface ∂Q and we will consider a vector field $J\boldsymbol{v}$ within this volume. Here, J is an intensive property of the fluid and $\boldsymbol{v}$ is the fluid velocities. According to the Gauss-Ostrogradsky (also known as divergence) theorem the net production/consumption of J by all sources and all sinks (which can be deemed sources with a negative contribution to J) within Q is equal to the net flow out of the region. Or, in mathematical terms, the divergence of the vector field, which reflects the strength of the sources and sinks, adds up to the outward flow through the surface:

$$\int_Q \nabla \cdot (J\boldsymbol{v})\,\mathrm{d}V = \oint_{\partial Q} J\boldsymbol{v} \cdot \boldsymbol{n}\mathrm{d}S, \tag{1}$$

where $\boldsymbol{n}$ is the vector normal to the surface. We couple this equation to Reynolds' Transport Theorem, which defines the change in a given property with respect to time as the result of outward flow from the region and the production/consumption by sources and sinks, h, within it:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_Q J\mathrm{d}V = -\oint_{\partial Q} J\boldsymbol{v} \cdot \boldsymbol{n}\mathrm{d}S - \int_Q h\mathrm{d}V \tag{2}$$

Using the divergence theorem we can convert the surface integral on the right hand side of Eq. 2 into a volume integral according to Eq. 1. Then, applying Leibniz's rule we can change the order

of differentiation and integration and bring the time derivative inside the integral. Rearranging the equation and combining the three volume integral yields:

$$\int_Q \left(\frac{\partial J}{\partial t} + \nabla \cdot (J\boldsymbol{v}) + h \right) \mathrm{d}V = 0 \tag{3}$$

This equation must hold for any closed surface, which is only possible when the integrand itself equals zero:

$$\frac{\partial J}{\partial t} + \nabla \cdot (J\boldsymbol{v}) + h = 0, \tag{4}$$

which is the generalized form of the continuity equation for any scalar fluid property.

2. Mass conservation

To impose mass conservation we use the fluid mass density as J and set $h = 0$, since there are no sources or sinks of mass, which physically corresponds to a system where no chemical reactions take place:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\boldsymbol{v}) = 0 \tag{5}$$

A (virtually) incompressible fluid like water does not change its density with time, nor are there any pressure gradients. Hence, the first term in the last equation is equal to zero and if we divide by ρ, we obtain the *mass continuity equation*:

$$\nabla \cdot \boldsymbol{v} = 0 \tag{6}$$

3. Momentum conservation

The continuity equation can be written not only for a scalar but also for a vector, such as the momentum. We start by substituting $J = \rho\boldsymbol{v}$ in Eq. 4:

$$\frac{\partial \rho\boldsymbol{v}}{\partial t} + \nabla \cdot (\rho\boldsymbol{v}\boldsymbol{v}) + \boldsymbol{h} = 0 \tag{7}$$

Here the source/sink term is a vector due to the tensor rank of the equation – all other terms in it are vector quantities. We write the time derivative and the divergence of the velocity dyad explicitly:

$$\rho\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v}\frac{\partial \rho}{\partial t} + \boldsymbol{v}\boldsymbol{v} \cdot \nabla\rho + \rho\boldsymbol{v} \cdot \nabla\boldsymbol{v} + \rho\boldsymbol{v}\nabla \cdot \boldsymbol{v} + \boldsymbol{h} = 0 \tag{8}$$

The third and fifth term can be combined into $\nabla \cdot (\rho\boldsymbol{v})$. Grouping these two terms with the time derivative of the density yields:

$$\boldsymbol{v}\underbrace{\left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\boldsymbol{v}) \right)}_{=0} + \rho \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla\boldsymbol{v} \right) + \boldsymbol{h} = 0 \tag{9}$$

We have already proven in the previous subsection the first term is zero for an incompressible fluid . The second term is known as the *material derivative* D/Dt of $\boldsymbol{v}$ and is nothing more than the total derivative of $\boldsymbol{v}\,(t, x, y, z)$ with respect to time:

$$\begin{aligned} \frac{D}{Dt}\boldsymbol{v}\,(t, x, y, z) &= \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{v}\,(t, x, y, z) \\ &= \frac{\partial \boldsymbol{v}}{\partial t} + \frac{\partial \boldsymbol{v}}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial \boldsymbol{v}}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial \boldsymbol{v}}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}t} \\ &= \frac{\partial \boldsymbol{v}}{\partial t} + v_x\frac{\partial \boldsymbol{v}}{\partial x} + v_y\frac{\partial \boldsymbol{v}}{\partial y} + v_z\frac{\partial \boldsymbol{v}}{\partial z} \\ &= \frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla\boldsymbol{v} \end{aligned} \tag{10}$$

We can now write Eq. 9 in a concise manner:

$$\rho \frac{D\boldsymbol{v}}{Dt} + \boldsymbol{h} = 0 \tag{11}$$

Subtracting $\boldsymbol{h}$ from both sides of the equation yields the body force equivalent of Newton's second law $m\boldsymbol{g} = \boldsymbol{F}$:

$$\rho \frac{D\boldsymbol{v}}{Dt} = \boldsymbol{h} \tag{12}$$

We omit the negative sign because the definition of sources and sinks, hence, their sign, is arbitrary.

4. Cauchy Stress Tensor

It is generally accepted that the sources and sinks can be classified into two main groups: surface and body forces. The former are given by the divergence of the Cauchy stress tensor, $\nabla \cdot \boldsymbol{\sigma}$, that is, the stress flux through the surface of a given control volume. On the other hand, the most common body force acting on the fluid particles is gravity. Others include electromagnetic, centrifugal or Coriolis forces:

$$\rho \frac{D\boldsymbol{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} \tag{13}$$

The stress tensor is usually split into two contributions: pressure forces which change the volume of the system and stresses which change its shape and position. These two contributions are mathematically represented by two tensors, the volumetric stress tensor, which is a 3-by-3 unity matrix multiplied by the negative pressure $-p$, and the stress deviator tensor $\boldsymbol{T}$:

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xy} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{pmatrix} = \begin{pmatrix} -p & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & -p \end{pmatrix} + \begin{pmatrix} \sigma_{xx}+p & \tau_{xy} & \tau_{xy} \\ \tau_{yx} & \sigma_{yy}+p & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz}+p \end{pmatrix} = -p\boldsymbol{I} + \boldsymbol{T} \tag{14}$$

Plugging this equation's final result in Eq. 13, we obtain the most general form of the Navier-Stokes equation:

$$\rho \frac{D\boldsymbol{v}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{T} + \boldsymbol{f} \tag{15}$$

To write down the deviatoric stress tensor explicitly we first need to introduce the concept of dynamic viscosity η which is the fluid's ability to dissipate momentum and relates the sheer stress and strain rate:

$$\tau_{ij} = \eta \frac{\mathrm{d}v_i}{\mathrm{d}x_j}, \tag{16}$$

where v_i is the i^{th} component of the velocity vector and x_j is a coordinate (in Cartesian coordinates i and j are x, y and z). If the viscosity is constant regardless of the strain rate, that is, if the sheer stress is linear in the strain rate, then we speak of a *Newtonian fluid*. For an incompressible Newtonian fluid the components of $\boldsymbol{T}$ are given by:

$$\boldsymbol{T}_{i,j} = \eta \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \tag{17}$$

We write down the second term on the right-hand side in Eq. 15 explicitly by making use of the short-hand notation for a partial derivative $\partial^n / \partial x^n = \partial_{x^n}$:

$$\nabla \cdot \boldsymbol{T} = \eta \nabla \cdot \begin{pmatrix} 2\partial_x v_x & \partial_y v_x + \partial_x v_y & \partial_z v_x + \partial_x v_z \\ \partial_x v_y + \partial_y v_x & 2\partial_y v_y & \partial_z v_y + \partial_y v_z \\ \partial_x v_z + \partial_z v_x & \partial_y v_z + \partial_z v_y & 2\partial_z v_z \end{pmatrix} \tag{18}$$

For simplicity we focus on the $\boldsymbol{T}_{x.j}$ component of $\boldsymbol{T}$:

$$\begin{aligned}\nabla \cdot \boldsymbol{T}_{x,j} &= \eta \begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix} \cdot \begin{pmatrix} 2\partial_x v_x \\ \partial_y v_x + \partial_x v_y \\ \partial_z v_x + \partial_x v_z \end{pmatrix} \\ &= \eta \left(2\partial_{xx} v_x + \partial_{yy} v_x + \partial_{xy} v_y + \partial_{zz} v_x + \partial_{xz} v_x\right) \\ &= \eta \nabla^2 v_x + \eta \partial_x \underbrace{\nabla \cdot \boldsymbol{v}}_{=0}\end{aligned} \tag{19}$$

In the second step we group the second-order pure partial derivatives into the Laplacian of v_x. The remaining terms can be represented as the first derivative of the velocity's divergence with respect to x. From mass conservation we know $\mathbf{div}\,(\boldsymbol{v})$ is zero. Carrying out identical calculation for the other two components of $\boldsymbol{T}$ results in a simple expression for $\mathbf{div}\,(\boldsymbol{T})$:

$$\nabla \cdot \boldsymbol{T} = \eta \nabla^2 \boldsymbol{v} \tag{20}$$

5. Dimensionless Form of NSE and Reynolds Number

We are now able to write down the final form of the NSE for an incompressible isotropic Newtonian fluid:

$$\frac{D\boldsymbol{v}}{Dt} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{v} + \boldsymbol{g} \tag{21}$$

We have divided through by the density, thus introducing the kinematic viscosity $\nu = \eta\rho^{-1}$. In this form the NSE depends on many different parameters of the system – density, viscosity, length and time scales etc. To generalize even further we convert all external parameter (that is, those which can be altered without changing the fluid itself) to their dimensionless counterparts (denoted by tildes) via a characteristic length scale L and a characteristic velocity u:

$$\boldsymbol{v} = u\tilde{\boldsymbol{v}} \qquad t = \frac{L}{u}\tilde{t} \qquad \frac{\partial}{\partial x} = \frac{1}{L}\frac{\partial}{\partial \tilde{x}} \qquad p = \rho u^2 \tilde{p} \qquad \boldsymbol{g} = \frac{u^2}{L}\tilde{\boldsymbol{g}} \tag{22}$$

Making all substitutions and multiplying the resulting equation by Lu^{-2} results in the dimensionless form of the NSE (for convenience we drop all tildes):

$$\frac{D\boldsymbol{v}}{Dt} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2 \boldsymbol{v} + \boldsymbol{g} \tag{23}$$

The dimensionless number $\mathrm{Re} = uL\nu^{-1}$ is known the *Reynolds number* and is defined as the ratio of the inertial forces to the viscous drag. This number characterizes the motion of a particle in the fluid: if $\mathrm{Re} \gg 1$ the flow is turbulent and inertial effects play a major role, while for $\mathrm{Re} \ll 1$ viscous drag is the dominating force acting on the suspended particle. The importance of Re is evident in the context of the dynamic similarity principle, which states that two systems with similar geometry (identical shape, different sizes) and identical Reynolds and Euler numbers behave identically. In other words, if the parameter choice in a simulation yields Re and Eu which are close if not equal to those for the experimental system, then the simulation can accurately predict the behaviour of the system for a wide variety of parameters.

6. Flow Between Parallel Plates

In this subsection we will derive the steady-state velocity profile expected for a fluid flowing between two parallel plates driven by a bodyforce, which is the system we will be simulating. Though the experimental system uses a pressure drop to set the fluid in motion, it has been shown [18] that a gravity-type force is an adequate substitute and has fewer pitfalls.

We start our derivation from Eq. 21 and we eliminate some of the terms. Since we are interested in the steady-state regime of the flow, the velocity field does not change in time, *i.e.*, the material derivative of the velocity is zero. There is no pressure drop in our system, hence we also neglect the first term on the right hand side. Furthermore, for z-walls and periodic BC in the x- and y-directions, only the flow component of the velocity is non-zero and it depends solely on the z-coordinate: $v_x(z) \neq 0$, $v_y = v_z = 0$. Finally, at the boundaries $z = 0$ and $z = L_z$, the velocity is always zero: $v_x(z = 0, t) = v_x(z = L_z, t) = 0$. By putting these arguments together we write the equation of motion and the two boundary condition equations:

$$\begin{aligned} \nu \frac{\partial^2 v_x(z)}{\partial z^2} + g_x &= 0 \\ v_x(z = 0) &= 0 \\ v_x(z = L_z) &= 0 \end{aligned} \tag{24}$$

Separating variables in the first equation and integrating twice gives:

$$v_x(z) = -\frac{g_x}{\nu}\frac{z^2}{2} + zC_1 + C_2, \tag{25}$$

where C_1 and C_2 are integration constant which we determine from the BCs. Setting $z = 0$ we deduce $C_2 = 0$. Using the second BC and solving for C_1 yields in:

$$C_1 = \frac{g_x}{\nu}\frac{L_z}{2} \tag{26}$$

Combining this result with the first line of Eq. 24 and rearranging, we obtain the final form of the velocity profile:

$$v_x = \frac{g_x}{2\nu} z(L_z - z) \tag{27}$$

The resulting parabola is known as a *Poiseuille profile* and it is typical for flow in a slit and in a tube. The maximum velocity is achieved in the mid-plane of the channel:

$$v_{\max} = \frac{g_x L_z^2}{8\nu} \tag{28}$$

and it is this velocity which is considered characteristic of the flow and is used to determine the Reynolds number. The characteristic length scale, on the other hand, is the depth of the channel. Then, the Reynolds number for a plane Poiseuille flow can be explicitly written:

$$\mathrm{Re}_{\mathrm{Pois}} = \frac{g_x L_z^3}{8\nu^2} \tag{29}$$

The importance of the last three results becomes clear when we put them in the context of our upcoming task. We will use Eq. 29 when choosing the magnitude of the body-force used in the simulations and eq. 27 and 28 to validate the fluid flow generated by LAMMPS in a simple manner.

III. Methods

Lattice Boltzmann Method (LBM)

1. Algorithm

Stemming from Lattice Gas Automata [19], the Lattice Boltzmann method aims at solving the Boltzmann equation on a Cartesian grid. Each grid point represents a fluid pocket and is characterized by

the one-particle distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$, which gives the probability of finding a particle with velocity $\boldsymbol{v}$ at position $\boldsymbol{x}$ at time t. In LBM both space and time are discretized which leads to a finite number of possible velocities. Depending on the dimensionality of the system and the model chosen the total number of accessible velocities varies. A common implementation of LBM for three-dimensional fluids is the D3Q19, where 19 is the total number of velocities. In this implementation a particle is positioned in the center of a cube and may move to the center of one of the faces (6), the center of one of the edges (12) or remain where it is. The probabilities for these movements are 1/18, 1/36 or 1/3, respectively. A high symmetry in particle motion is a required to recover the proper hydrodynamic behaviour, hence, the large number of possible movement directions.

The algorithm consists of two steps – streaming, during which particles positions are updated, and collision, which ensures fluid relaxation to the local equilibrium distribution (for simplicity we assume a time increment of 1):

$$f(\boldsymbol{x}+\boldsymbol{v}, \boldsymbol{v}, t+1) - f(\boldsymbol{x}, \boldsymbol{v}, t) + \boldsymbol{F}(\boldsymbol{v}) = \frac{1}{\tau}\left(f^{\mathrm{eq}}(\boldsymbol{x}, \boldsymbol{v}, t) - f(\boldsymbol{x}, \boldsymbol{v}, t)\right) \tag{30}$$

Here, $\boldsymbol{F}(\boldsymbol{v})$ is a forcing term, which adds a body-force to each particle, thus inducing flow. In the BGK model [20], the relaxation to local equilibrium $f^{\mathrm{eq}}(\boldsymbol{x}, \boldsymbol{v}, t)$ is a single-step process characterised by the characteristic time $\tau = (1+6\nu)/2$, with ν being the the kinematic viscosity of the fluid.

2. Dimensionless Numbers and Parameter Choice

One of the crucial features of colloids in Q2D flow is the lubrication layers separating the colloid from the walls. They mediate the friction between the colloid and the plates, thus, considerably reducing its velocity with respect to the far field fluid. This ultimately leads to the dipolar shape of the flow velocity profile in the xy-plane around the dumbbell. Acknowledging the importance of these layers, start building a system in which the lubrication layers are fully resolved – their thickness δ in grid edge lengths should be at least 4. Using the experimental system geometry [21], we set $\delta = 5\,a$ and write down the following ratios:

$$\frac{\delta}{L_z} = \frac{\text{lubrication layer thickness}}{\text{channel depth}} \approx 0.06 \quad \Rightarrow L_z = 70\,a \tag{31}$$

$$\frac{R}{L_z} = \frac{\text{disc radius of dumbbell}}{\text{channel depth}} \approx 0.80 \quad \Rightarrow R = 55\,a \tag{32}$$

$$\frac{L_y}{R} = \frac{\text{channel width}}{\text{disc radius of dumbbell}} = 12 \quad \Rightarrow L_y = 660\,a \tag{33}$$

$$\frac{L_x}{L_y} = \frac{\text{channel length}}{\text{channel width}} = 2 \quad \Rightarrow L_x = 1320\,a \tag{34}$$

These parameters yield a system of $L_x \times L_y \times L_z = 1320 \times 660 \times 70$, dimensions we employ for the first few simulations we carry out. Later in our study we switch to a simulation box $L_x \times L_y \times L_z = 800 \times 800 \times 70$ because the velocity field distortion, caused by the dumbbell in the flow direction, does not span more than 300 grid points upstream and downstream from the colloid. Therefore, we can truncate the simulation box in the x-direction to reduce computational time.

In this implementation of LBM, the colloid is represented by a mesh of N_{mesh} point particles which are held together by a rigid potential. The mesh is coupled to the fluid by a four-point interpolation scheme based on the immersed boundary method proposed by Lai and Peskin [22], which approximates the momentum transfer between the fluid and the colloid. Each dumbbell consists of two discs of thickness $h = H - 2\delta = 60$, whose centers are separated by $s = 3.3\,R$ grid points. The disks themselves are connected by a shaft which is $4R/3$ wide and has the same thickness as the discs. For a symmetric dumbbell both discs have a radius of $R = 55\,a$. To demonstrate how asymmetry affects the trajectory of the dumbbell, we also investigate a system where one of the disc has a diameter of 121 grid points ($\tilde{R} = 1.1\,R$).

A grid spacing of $a = 4\,\mu\text{m}$ and a timestep of $t_{\text{MD}} = 5\,\mu\text{s}$ are chosen (LAMMPS' LBM package supports calculations with real units), because these parameters yield a numerically stable simulation with a sufficiently large timestep – combinations of smaller a and different t_{MD} have caused the simulation to terminate. There is one exception: in our initial simulation, when we use a large box (1320 by 660 by 70), we chose $a = 3\,\mu\text{m}$ and $t_{\text{MD}} = 3\,\mu\text{s}$. In all simulations the colloid mesh density, given by the ratio of the surface area of the colloid to the number of mesh points forming it, is kept smaller than the grid density, a^2.

The last parameter we need to define is the driving acceleration, applied to every LBM fluid particle. We can extract its value for a given Reynolds number via Eq. 29, as long as we know the characteristic length of the flow and the fluid's kinematic viscosity. We set the fluid mass density and dynamic viscosity to their physical counterparts known for water($\rho = 1.0\,\text{pg}\,\mu\text{m}^{-3} = 1000\,\text{kg}\,\text{m}^{-3}$ and $\eta = 1.0\,\text{pg}/(\mu\text{m}\,\mu\text{s}) = 1.0\,\text{mPa}\,\text{s}$) and solve for the driving acceleration (we change L_z to aL_z to take into account the lattice spacing):

$$g\left(\text{Re} = 0.1\right) = \frac{8\nu^2}{a^3 L_z^3}\text{Re} \approx 36\,\text{mm}\,\text{s}^{-2} \tag{35}$$

We also carry out some simulations with a smaller grid spacing – in these cases we choose a driving acceleration $g\left(a = 3\,\mu\text{m}\right) = 250\,\text{mm}\,\text{s}^{-2}$, which corresponds to $\text{Re} \approx 0.3$.

B. Stochastic Rotational Dynamics (SRD)

1. Algorithm

An MPCD fluid consists of N_{t} point particles with mass m_{f} placed in a simulation box with dimensions $L_x,\ L_y$ and L_z. The temperature of the system is T. Each particle is characterized by its position $\boldsymbol{r}_i$, velocity $\boldsymbol{v}_i$, mass m_{f}. The subscript i denotes the particle number and runs from 1 to N_{t}. The particle mass is commonly set to unity. The MPCD algorithm consists of N_{coll} consecutive streaming steps and a collision step. During the former each particle undergoes a ballistic move over a timestep t_{MD} and its position is updated:

$$\boldsymbol{r}_i\left(t + t_{\text{MD}}\right) = \boldsymbol{r}_i\left(t\right) + t_{\text{MD}}\boldsymbol{v}_i\left(t\right) \tag{36}$$

The streaming step is repeated N_{coll} times before a collision takes place, *i.e* particles collide with each other every $t_{\text{c}} = N_{\text{coll}}t_{\text{MD}}$ timesteps. Then the simulation box is split into cubic collision cells with edge length a. The resulting average number of particles per cell, $\gamma = N_{\text{t}}a^3 L_x^{-1}L_y^{-1}L_z^{-1}$, depends on the fineness of the mesh. Usually, an edge length $a = 1$ is chosen, thus, γ matches the total particle density. The collisions between particles within a given cell are performed via a stochastic process, which resets their velocities while conserving the total momentum of the cell. The most commonly-employed collision rule is the SRD rule, introduced by [9]. Its name stems from the stochastic rotation of the relative velocities of the particles $\boldsymbol{\delta v}_i\left(t\right) = \boldsymbol{v}_i\left(t\right) - \boldsymbol{v}_{\text{c.m.}}\left(t\right)$ with respect to the center-of-mass velocity of the collision cell $\boldsymbol{v}_{\text{c.m.}}\left(t\right) = N_{\text{c.c.}}^{-1}\sum_{k=1}^{N_{\text{c.c.}}}\boldsymbol{v}_k\left(t\right)$. Here $N_{\text{c.c.}}$ is the number of particles in the cell of interest. Implementation of the collision is achieved via the matrix $\mathbf{R}$, representing the rotation by a fixed angle α with respect to a randomly-chosen axis $\hat{\mathbf{R}}$, which is applied to $\boldsymbol{\delta v}_i$:

$$\boldsymbol{v}_i\left(t + t_{\text{MD}}\right) = \boldsymbol{v}_{\text{c.m.}}\left(t\right) + \mathbf{R}\cdot\boldsymbol{\delta v}_i\left(t\right), \tag{37}$$

Written out explicitly, Eq. 36 reads (Appendix of [18]):

$$\boldsymbol{v}_i\left(t + t_{\text{MD}}\right) = \boldsymbol{v}_{\text{c.m.}}\left(t\right) + \boldsymbol{\delta v}_{i,\perp}\left(t\right)\cos\alpha + \left(\boldsymbol{\delta v}_{i,\perp}\left(t\right)\times\hat{\mathbf{R}}\right)\sin\alpha + \boldsymbol{\delta v}_{i,\parallel}\left(t\right), \tag{38}$$

where $\perp$ and $\parallel$ denote the components of $\boldsymbol{\delta v}_i$ which are perpendicular and parallel to $\hat{\mathbf{R}}$. It has been proven that an H-theorem exists for the algorithm, the single-particle velocity distributions follow a Maxwell-Boltzmann distribution and the fluid itself obeys an ideal gas law [9][23].

Although its equation of state is that of an ideal gas, the MPCD fluid can act as either a liquid or a gas, depending on the system parameters chosen. To illustrate this, we introduce the dimensionless mean free path λ as the ratio of intercollision distance traveled by the particles to the characteristic length scale of our system, in this case the collision cell size:

$$\lambda = t_{\mathrm{c}} \sqrt{\frac{k_{\mathrm{B}} T}{m_{\mathrm{f}}}} \frac{1}{a} = \frac{t_{\mathrm{c}}}{t_0}, \tag{39}$$

where $t_0 = a\sqrt{m_{\mathrm{f}} (k_{\mathrm{B}} T)}^{-1}$ is the characteristic timescale of the fluid. A short λ (short collision times) corresponds to a highly-correlated system, which behaves like a liquid. In contrast, large mean free paths are typical for gaseous systems in which particles rarely collide with each other. Another parameter defining the SRD fluid as either a gas or a liquid is the rotation angle, which characterizes the strength of the interparticle interaction – the bigger the angle, the stronger the interaction. The importance of the rotation angle can be understood intuitively if we set it to zero: the particle velocities are not reset during the collision step, hence, particles do not 'see' each other, which is the ideal gas case.

Simulating a highly-correlated fluid like water requires an additional step to the standard streaming and collision ones. Since it takes t_{c}^{-1} collisions for a particle to travel a distance a, it participates in several consecutive collisions with one and the same group of particles, which violates the Galilean invariance principle. To restore invariance, the origin of the collision grid is randomly-shifted before any rotation has been applied. The shifting parameter is taken from a uniform distribution with bounds $[-a/2; a/2]$. Thus, the neighbors of each particle are randomized before every collision without changing any particle positions.

Additionally to their thermal motion, fluid particles exhibit directed movement in alignment with the flow. There are several ways one can achieve this forced advection, the most common of which are applying either a pressure drop across the simulation box or a gravity-type body force $\boldsymbol{F} = m_{\mathrm{f}} \boldsymbol{g}$ acting on each particle. Given the pressure gradient used in the experimental system, the former method may seem the natural choice. However, [18], who investigated forced flow of a liquid between two parallel plates, have proven the method leads to fluctuations in particle density around the inlet and outlet, rendering it impractical for our goals. In the same work the expected Poiseuille profile was obtained by implementing the latter method. What makes the bodyforce method particularly appealing is the straightforward manner of its implementation – one simply needs to modify the rule governing position updates so it takes into account the driving acceleration:

$$\boldsymbol{r}_i (t + t_{\mathrm{MD}}) = \boldsymbol{r}_i (t) + t_{\mathrm{MD}} \boldsymbol{v}_i (t) + \boldsymbol{g} \frac{t_{\mathrm{MD}}^2}{2} \tag{40}$$

and add another rule which modifies the velocities of the particle each MD timestep:

$$\boldsymbol{v}_i (t + t_{\mathrm{MD}}) = \boldsymbol{v}_i (t) + \boldsymbol{g} t_{\mathrm{MD}} \tag{41}$$

No alteration of the collision algorithm is needed, since it only randomizes the velocities of the particles, without changing their positions or $|\boldsymbol{v}|$.

The last problem one needs to consider when implementing an SRD algorithm is the fluid-wall coupling in both its aspects – the collision rule and the filling of collision bins near a wall. The former deals with how the particles 'see' the walls and how they bounce off them, and is present in both fluid- and gas-like systems. Conversely, the collision bin filling issue is particularly relevant to SRD fluids. After a random grid shift both walls will be slicing through collision cells which they would otherwise border, effectively leading to partial filling of these cells. Ameliorating this problem requires filling these cells with virtual particles prior to the velocity randomization.

The wall collision issue has been studied extensively and it is commonly accepted that the simple bounce-back rule is sufficient to recover no-slip boundary conditions [24][25]. When a particle enters the wall, it is placed back at the point of collision with the wall $\boldsymbol{r}_{i,\mathrm{wall}} (t + t^\star)$ and the time lapse between the

beginning of the current streaming and the wall collision is calculated $t^\star = t_{\mathrm{MD}} - \boldsymbol{r}_{i,\mathrm{wall}}\left(t + t^\star\right)/\boldsymbol{v}_i\left(t\right)$. The velocity of the particle is updated by reversing the present one $\boldsymbol{v}_{i,\mathrm{new}} = -\boldsymbol{v}_i\left(t\right)$, the particle is streamed for the remainder of the current step $t_{\mathrm{MD}} - t^\star$ and its velocity is not updated until the next collision step or the next wall impact. An alternative to the bounce-back rule is the stochastic reflection, which has been suggested by Inoue *et al.*. A study by Bolintineanu and co-workers [26], however, reported some residual slip.

These researchers also looked into the problem of partial bin filling and concluded that prior knowledge about the nature of the flow is needed to fully remove slip at the wall – the virtual particles should be assigned velocities which comply with the parabolic Poiseuille profile. Between the walls the velocity is positive in the flow direction and within them it is negative. In order to recover no-slip boundaries one needs to choose the velocity of the virtual particle from a Gaussian distribution centered around the negative velocity which the parabola assigns to this particle position.

2. Dimensionless Numbers and Parameter Choice

Similarly to LBM, in SRD one also need to account for the finite thickness of the lubrication layers separating the dumbbell from the upper and lower wall. However, in SRD a fully resolved lubrication layer requires lower grid fineness - at least 2 a are required for two spheres in close proximity. Taking this into account, we calculate the dimensions of the simulation box in a fashion similar to the one described in the section about LBM and choose $L_x \times L_y \times L_z = 350 \times 350 \times 35$.

Since SRD fluid parameters cannot be directly mapped to fluid properties, we are unable to use physical quantities as we did in the previous section. Instead, we try to match the dimensionless numbers characterizing the fluid and the flow, namely, the Schmidt and Reynolds numbers, to the one known for the experimental system.

The Schmidt number is the ratio of collision momentum transport to kinetic momentum transport: $\mathrm{Sc} = \nu D_0^{-1}$, with D_0 being the self-diffusion coefficient of the fluid. Sc determines if our SRD fluid has gas- or liquid-like properties – low values indicate momentum is transferred mainly through diffusion ($\mathrm{Sc} \lesssim 1$), while $\mathrm{Sc} \gg 1$ points to collisional momentum transport, typical for liquids.

Ihle and Kroll [23] have derived an expression for $D_0\left(\gamma, \alpha, \lambda\right)$, a result confirmed by Ripoll *et al.* [27]:

$$D_0 = \lambda \frac{a^2}{t_0}\left(\frac{3}{2 - 2\cos\alpha}\frac{\gamma}{\gamma - 1} - \frac{1}{2}\right) = \lambda \frac{a^2}{t_0}\tilde{D}_0. \tag{42}$$

The rotation angle α is kept fixed and usually either $\pi/2$ or $3\pi/4$ are used [15, 16, 28–30]. Inspecting Eq. 41, one notices the governing variable is the mean free path, since the term in parentheses is on the order of unity, regardless of what values are chosen for γ and α (Fig 1a). Therefore, it can be concluded:

$$D_0 \simeq \lambda \frac{a^2}{t_0} \tag{43}$$

The analytical expression for the kinematic viscosity derived by Kikuchi *et al.*[31] recognizes two terms contributing to ν, a kinetic part:

$$\nu_{\mathrm{kin}} = \frac{a^2}{t_0}\frac{\lambda}{3}\left(\frac{1}{1 - \gamma^{-1} + \mathrm{e}^{-(\gamma + \ln\gamma)}}\frac{15}{4 - 2\cos\alpha - 2\cos 2\alpha} - \frac{3}{2}\right) = \frac{\lambda}{3}\frac{a^2}{t_0}f_{\mathrm{kin}} \tag{44}$$

and a collisional one:

$$\nu_{\mathrm{coll}} = \frac{a^2}{t_0}\frac{1}{18\lambda}\left(1 - \cos\alpha\right)\left(1 - \gamma^{-1} + \mathrm{e}^{-(\gamma + \ln\gamma)}\right) = \frac{1}{18\lambda}\frac{a^2}{t_0}f_{\mathrm{coll}} \tag{45}$$

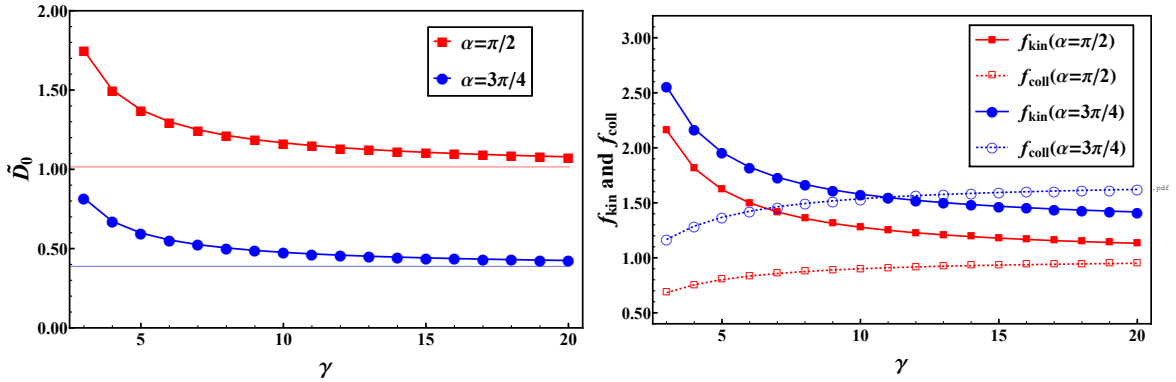
Both contributions are weakly dependent on the particle denisity and the rotation angle, as can be seen from Fig. 1b, and we are again left with a strong dependence solely on λ:

$$\nu_{\mathrm{kin}} \simeq \frac{\lambda}{3}\frac{a^2}{t_0} \quad \text{and} \quad \nu_{\mathrm{coll}} \simeq \frac{1}{18\lambda}\frac{a^2}{t_0} \tag{46}$$

For short mean free paths the collisional term dominates, in good agreement with our previous argument, *i.e.*, in liquids momentum transport is carried out predominantly by collisions. If $\lambda \gtrsim 1$, our system behaves like a gas and the main contribution to ν is the kinetic one.

Since we are interested in simulating a liquid, we can neglect the kinetic contribution and write down a fairly accurate estimate for the Schmidt number:

$$\mathrm{Sc} \simeq \frac{\nu_{\mathrm{coll}}}{D_0} \simeq \frac{1}{3} + \frac{1}{18\lambda^2} \tag{47}$$



(a) Weak dependence of D_0 on α and γ

(b) Weak dependence of ν_{kin} and ν_{coll} on α and γ

Figure 1: Weak dependence of different transport coefficients on the collision angle α and the average particles per cell γ. (a) The dimensionless self-diffusion coefficients $\tilde{D}_0$ (Eq. 42) for two collision angles hardly change for $\gamma \geq 5$ and differ by roughly 0.5. (b) The kinetic f_{kin} (44) and collision f_{coll} (45) contributions to the kinematic viscosity for two collision angles are compared. Regardless of α all contributions are roughly equal to unity for $\gamma \geq 5$

The last step in our analysis is to calculate the proper value for λ using the Schmidt number for water at standard conditions, $\mathrm{Sc} = 425$. The value we obtain, however, is impractical for our purposes because it is too small, $\lambda \simeq 10^{-2}$. To understand why this is so, we need to calculate the Reynolds number for such a system. We choose particle denisity of $\gamma = 5$ to reduce computation time, which scales with the number of particles, while avoiding rarefaction effects (as visible from Fig 1, ν and D_0 hardly change for higher densities). We use the rotation angle $\alpha = \pi/2$, which is hardwired into LAMMPS' SRD algorithm. With these parameters we calculate the kinematic viscosity of the fluid $\nu \simeq 5\, a^2/t_0$. If we now plug this value into the expression for Re (Eq. 28) while keeping in mind that Re can be 0.1 at most to eliminate inertial effects, we calculate the maximum possible velocity which complies with all criteria: $v_{\mathrm{max}} \simeq 0.02$.

Since the main goal of our study is to track the dumbbell for two box lengths, we can estimate the total runtime t_{run} in t_0 units:

$$t_{\mathrm{run}} = \frac{2L_x}{v_{\mathrm{max}}} = 34 \times 10^4\, t_0 \tag{48}$$

Here we assume the dumbbell velocity does not differ considerably from that of the surrounding fluid, but, as we will see, it is roughly half of the far field fluid velocity in LBM. Converting the runtime into timesteps requires us to divide it by the integration scheme timestep t_{MD}:

$$N_{\mathrm{steps}} = \frac{t_{\mathrm{run}}}{t_{\mathrm{MD}}} = N_{\mathrm{coll}} \frac{t_{\mathrm{run}}}{\lambda t_0} \approx 5 \times 10^6\, N_{\mathrm{coll}} \tag{49}$$

Even if the number of consecutive streaming steps is set to unity, a single simulation of a pure fluid between two walls would take 1 to 2 days on 240 CPU cores clocked at 3 GHz (rough estimates based

on preliminary data). A number of authors have used Sc ranging from 5 to 50. In the light of these studies we use Sc=20, but emphasize that $t_{\mathrm{run}}\,(\mathrm{Sc})$ poses an interesting issue and will be part of our future research.

Using Sc=20 and Eq. 47, we calculate $\lambda = 4 \times 10^{-2}\, a$ and $\nu \approx 1.2\, a^2/t_0$, correspond to a maximum velocity of $v_{\max} \approx 3 \times 10^{-3}\, a/t_0$. With these parameters we can easily compute the driving acceleration according to Eq. 29 $g \approx 30 \times 10^{-6}\, a/t_0^2$.

IV.Results

In this chapter we present the results we obtain with the two computational algorithms described in Chapter III - Lattice Boltzmann Method and Stochastic Rotational Dynamics. It is important to note that they both have shortcomings in the way they are implemented in LAMMPS – neither algorithm supports more than one set of walls, the SRD fluid cannot be coupled to a non-spherical object and terminates when a rigid body composed of spheres is used, LBM is numerically unstable and relatively slow, etc. All these issues prevent us from simulating a system similar to the experimental and will be discussed, alongside possible remedies, in the following chapter. Bearing this in mind, we present preliminary result, which, may not necessarily expanding our understanding of the system, but serve as basis for future research on the topic.

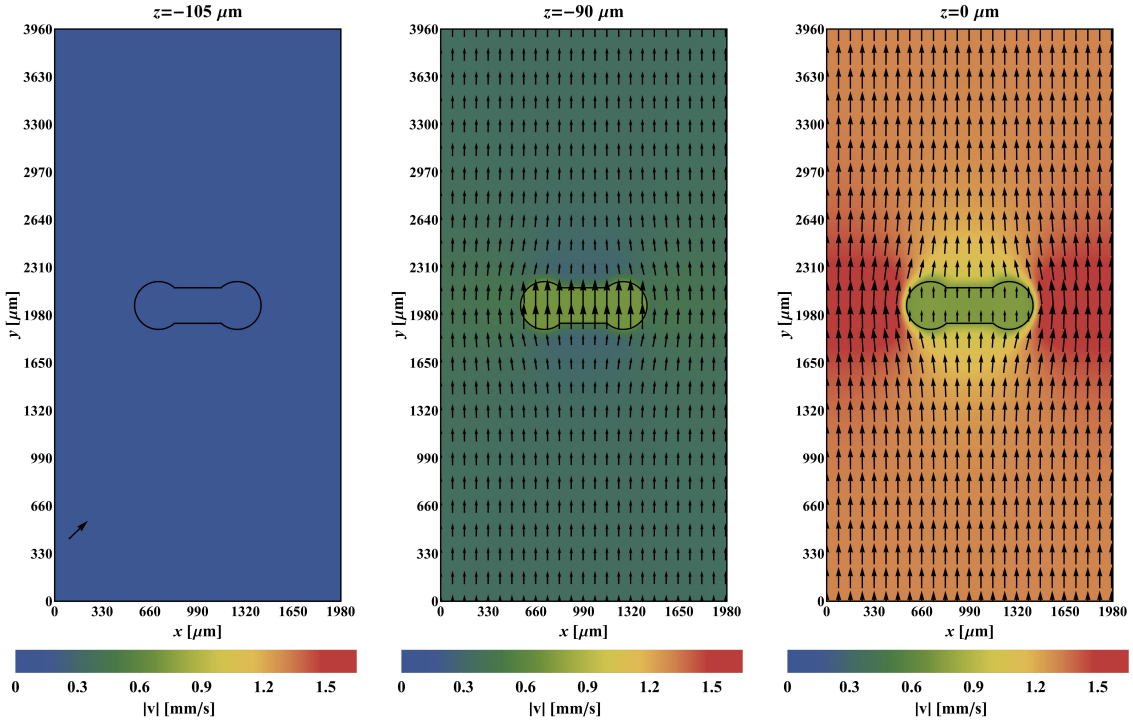
LBM

We start our investigation with a large system ($660\, a \times 1320\, a \times 70\, a$) with the following parameters $(a, t_{\mathrm{MD}}, g_y) = \left(3\,\mu\mathrm{m}, 3\,\mu\mathrm{s}, 250\,\mathrm{mm\,s^{-2}}\right)$. The colloid mesh consists of 246087 point particles of mass $m_{\mathrm{c}} = 550\,\mathrm{pg}$ and is generated with Gmsh [32]. The resulting colloid mesh density is $3.19\,\mu\mathrm{m}^{-2}$. The simulation is run for 1×10^5 timesteps, the first 3×10^4 of which being equilibration and the colloid is tracked for another 7×10^4 timesteps. Its shaft is perpendicular to the flow and it only moves in the y-direction (data not shown).

Fig. 2 compares the velocity fields in the simulation box at three depths - at the bottom plate ($-105\,\mu\mathrm{m}$), at the lower face of the dumbbell ($-90\,\mu\mathrm{m}$) and at the midplane ($0\,\mu\mathrm{m}$). The no-slip BC is recovered quite well, as we see from the velocity profile at $z = 0$. The vector at roughly $(50, 400)$ is most probably a post-processing error. As we go up, the fluid velocity around the dumbbell slowly increases (Supplementary video 1), in good agreement with the parabolic profile we expect. The fluid underneath the dumbbell is dragged forward by the colloid itself – this resembles Coeuette flow between parallel plates. As we move further up the velocity of the surrounding fluid increases and peaks in the mid-plane of the channel, where we observe a hydrodynamic dipole: the fluid flows faster around the dumbbell compared to the far field velocity. This effect is particularly visible for narrow (small x) channels and can be explained by the continuity equation: the dumbbell acts as a obstacle and effectively narrows the channel. Since the fluid is incompressible (at this Mach number) it needs to flow faster around the colloid thus conserving momentum along the channel.

From the figure it is also evident that the distortion of the velocity field does not span more than 300 grid points away from the center of mass of the dumbbell in the direction perpendicular to the shaft. In the parallel direction, however we notice a slow decay of the velocity to the far-field one. These observations enable us to make the simulation box smaller in the flow direction (from 1320 to 800 grid points) and make it slightly larger in the vorticity direction: from 660 to 800. These dimensions will be used in all following LBM simulations.

To make this dipolar nature of the flow even more evident we shift the frame of reference from the fluid to the dumbbell by subtracting its center-of-mass velocity from the fluid velocity field and again present several snapshots at different channel depths (Fig. 3). This shift is the reason why we observe 'flow' at the bottom plate Fig. (3a) – the stationary fluid in contact with the wall is 'seen' as a backflowing fluid by the colloid. This reasoning is valid even further up than the bottom face of the



(a) Bottom plate (b) Bottom face of dumbbell (c) Mid-plane

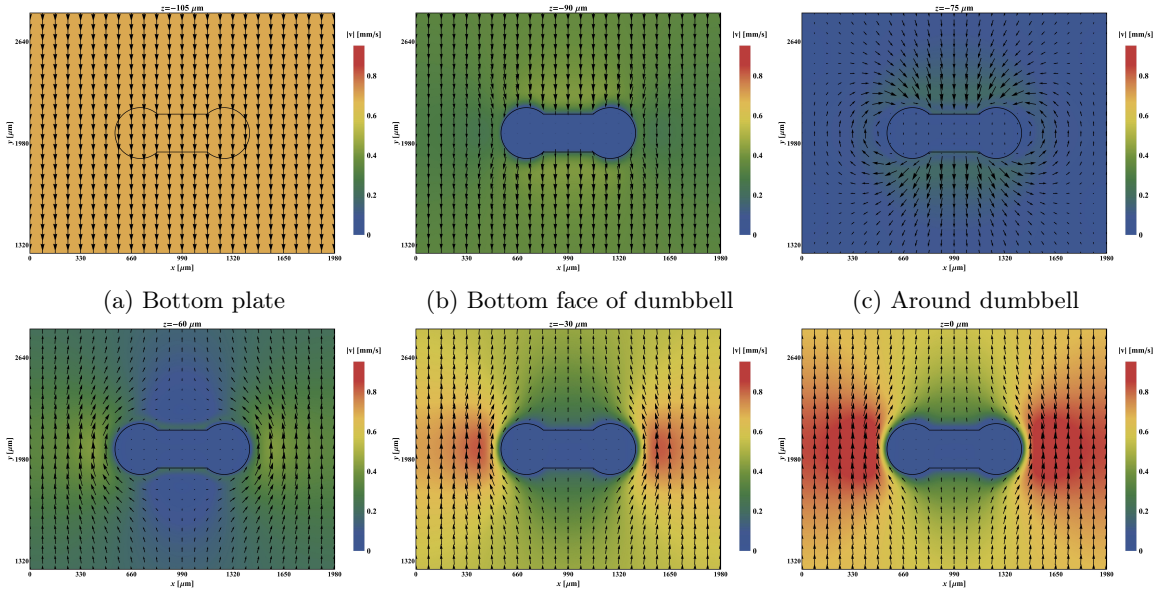
Figure 2: Velocity field in a channel at three different heights with respect to the bottom wall. (a) Bottom wall. (b) Bottom face of the dumbbell. (c) Mid-plane of the channel. The fluid velocity gradually increases from 0 at the wall to $v_{\max}$ in the mid-plane. The dumbbell introduces a flow distortion and moves with a lower velocity compared to the surrounding fluid.

dumbbell (Fig. 3b): around ten grid points from the bottom plate the fluid velocity near the dumbbell becomes equal to that of the colloid (Fig. 3c). As we move further up we observe a gradual increase in the fluid velocity surrounding the dumbbell (Fig 3d,e) which is most pronounced in the mid-plane of the channel.

After this preliminary simulation we switch to a system with a different size, lattice spacing, timestep and Re – $L_x \times L_y \times L_z = 800\,a \times 800\,a \times 70\,a$ and $(a, t_{\mathrm{MD}}, g_y) = \left(4\,\mu\mathrm{m}, 5\,\mu\mathrm{s}, 36\,\mathrm{mm\,s^{-2}}\right)$. We again keep the shaft perpendicular to the flow direction but this time run the simulation only for 2.5×10^4 timesteps because we are interested in the time evolution of $v_x\,(z)$ and the shape of the velocity profile after steady state flow has been achieved.

Fig. 4a presents the maximum velocity of the far field fluid as a function of time. As we can see, the simulation yields $v_{\max}$ which is lower than the one following from Eq. 28. This discrepancy can be attributed to the kinematic viscosity – the algorithm creates a fluid with ν, which is higher than the input one. To verify this assumption we calculate the kinematic viscosity corresponding to the long time plateau in $v_{\max}\,(t)$ and obtain $\nu_{\mathrm{fit}} = 1.066\,\mu\mathrm{m}^2\,\mu\mathrm{s}^{-1}$. The fitted kinematic viscosity also yields a parabola $v_y\,(z)$ which perfectly fits the simulation result Fig. 4b. This issue has been reported by [13] which calculate hydrodynamic radii for spheres which are slightly (on the order of 5 %) larger than the actual. A validation of our assumption can by made via the non-equilibrium Müller-Plathe method and will be done in a following work. Also, we should not rule out the increased channel resistance caused by the microparticle.

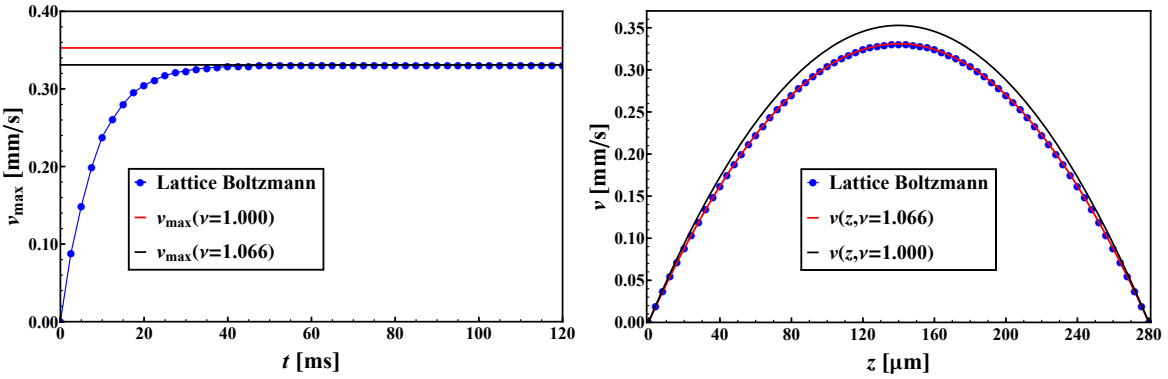
Our next attempt at recovering the motion of the dumbbell known from experiments involves tilting the dumbbell with respect to the x-axis, so it forms $\theta = \pi/4$ angle with the shaft. We use the same set



(a) Bottom plate (b) Bottom face of dumbbell (c) Around dumbbell

(d) Around dumbbell (e) Around dumbbell (f) Mid-plane

Figure 3: Velocity field in a channel at six different heights with respect to the bottom wall. The frame of reference is shifted from the fluid to the dumbbell, all velocities are offset by the center-of-mass velocity of the colloid.

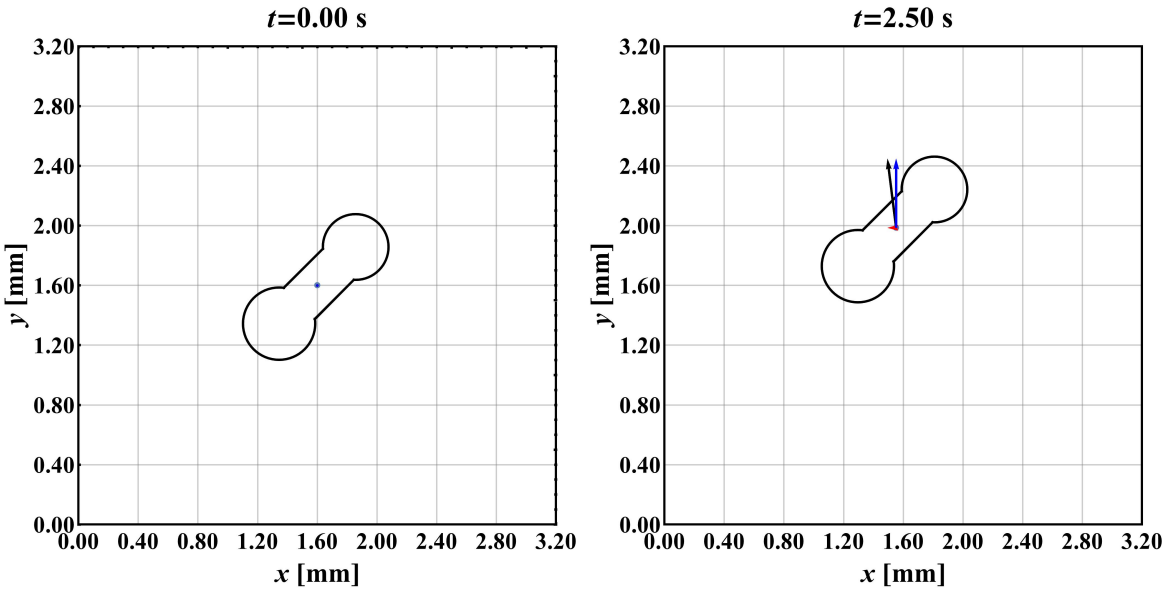


(a) $v_{\max}(t)$ (b) $v_y(z)$

Figure 4: (a) Time evolution of maximum velocity of the flow. The LBM algorithm generates a fluid with kinematic viscosity, higher than the one following from the input parameters. (b) Poiseuille profile $v_x(z)$ of the fluid. The simulation result corresponds to a fluid with higher viscosity, which confirms the result for v_{t}.

of parameters as in the previous case but we now run the simulation for 5×10^5 timesteps and track the dumbbell after the first 5×10^4. Introducing this asymmetry in the system induces a sideways motion as we expect but the effect is barely noticeable. As shown Fig. 5a and 5b the dumbbell drifts only slightly in the vorticity direction – this shift relative to the width of the simulation box is roughly 1 %. In contrast, over the same time, the dumbbell travels approximately 10 % of the box size in the flow direction.

Finally we turn our attention to an asymmetric dumbbell, *i.e.*, one in which the two discs differ in their radii. We expect to observe rotation which should orient the dumbbell's smaller disc downstream,



(a) Initial position and COM velocity of the dumbbell (b) Final position and COM velocity of the dumbbell

Figure 5: Motion of a symmetric dumbbell with $\theta = \pi/4$ simulated for $t_{\text{run}} = 2.5\,\text{s}$ with a timestep of $t_{\text{MD}} = 5\,\mu\text{s}$. The dumbbell drifts only slightly to the left.

a phenomenon already observed experimentally. We choose to make one of the discs larger by 10 %, thus, $R_1 = 220\,\mu\text{m}$ and $R_2 = 242\,\mu\text{m}$. With all other system parameters, θ, a, t_{MD} and g_y, kept the same, we run the simulation for 2×10^6 timesteps and track the colloid from $t = 0\,\text{s}$ onwards. The slow movement in the vorticity direction is still present but there is seemingly no rotation (Fig. 6). To check if the angular velocity ω_z is indeed zero we calculate $\theta(t)$ for both the symmetric and asymmetric dumbbell and present the results in Fig 7:

$$\theta(t) = \arccos\left(\frac{\boldsymbol{O}(t) \cdot \boldsymbol{e_x}}{|\boldsymbol{O}(t)|}\right), \tag{50}$$

Here, $\boldsymbol{O}(t)$ is the vector starting from the center of one disc (in the asymmetric case this is the bigger one) pointing to the other (inset of 7) and $\boldsymbol{e_x}$ is the unit vector in the x-direction. $\boldsymbol{O}(t)$ is oriented in such a way that its y-component is aligned with the flow.

The asymmetric dumbbell indeed rotates but with a small circular velocity on the order of $1\,\text{mrad}\,\text{s}^{-1}$. Though slow, the rotation is in the correct and physically sensible direction – the dumbbell is gradually orienting itself with its smaller disk downstream, as we expected. In contrast, the symmetric one does not only rotate 30 times slower, but it does so with a negative angular velocity. This negligible rotation is most probably due to small defects of the colloid mesh, which results in an uneven momentum transfer from the fluid to the colloid.

As a final word to this section, we shall discuss the efficiency of the algorithm in LAMMPS. The last system we described consisted of $N_{\text{g.t.}} = 800 \times 800 \times 70 = 44.8 \times 10^6$ grid points and we ran the simulation for 2 million timesteps. The calculation took approximately 5.5×10^5 seconds on 32 Intel Xeon® X5670 octa-core CPUs (or 256 cores in total) clocked at 2.93 GHz. In other words, the algorithm is capable of 0.64 million site updates per second per CPU core, which is 3 to 5 times slower than other algorithms written in C/C++ [33]. Keeping in mind that LAMMPS is an open-source multi-purpose simulation tool, there is always room for improvements, improvements which we intend to implement in future research.

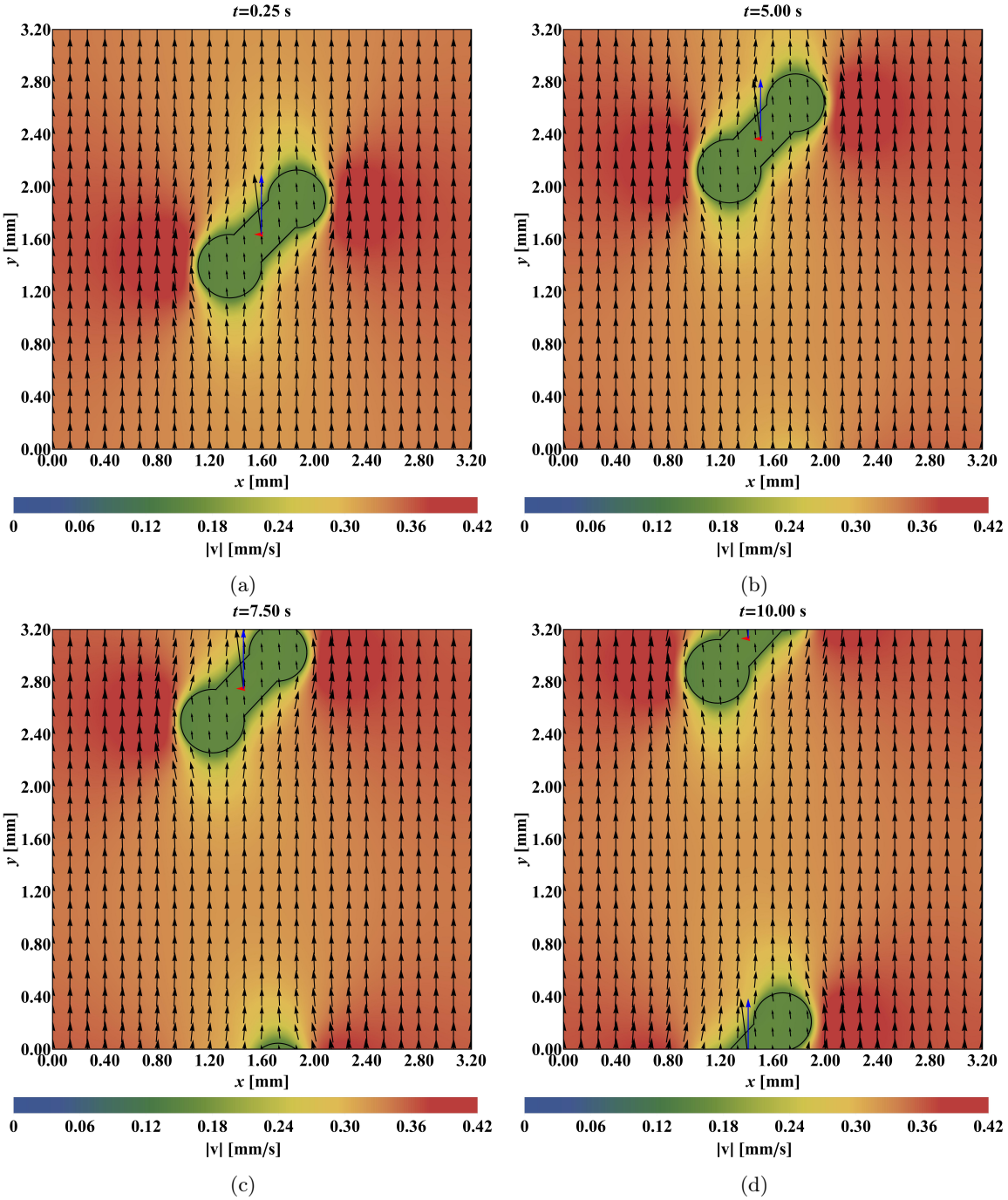


Figure 6: Motion of a symmetric dumbbell with $\theta = \pi/4$ simulated for $t_{\mathrm{run}} = 2.5\,\mathrm{s}$ with a timestep of $t_{\mathrm{MD}} = 5\,\mu\mathrm{s}$. The dumbbell drifts only slightly to the left.

SRD

We start of our investigation with the SRD algorithm using the knowledge we gained from our LBM simulations – we use a simulation box with dimensions $L_x \times L_y \times L_z = 350\,a \times 350\,a \times 35\,a$. The

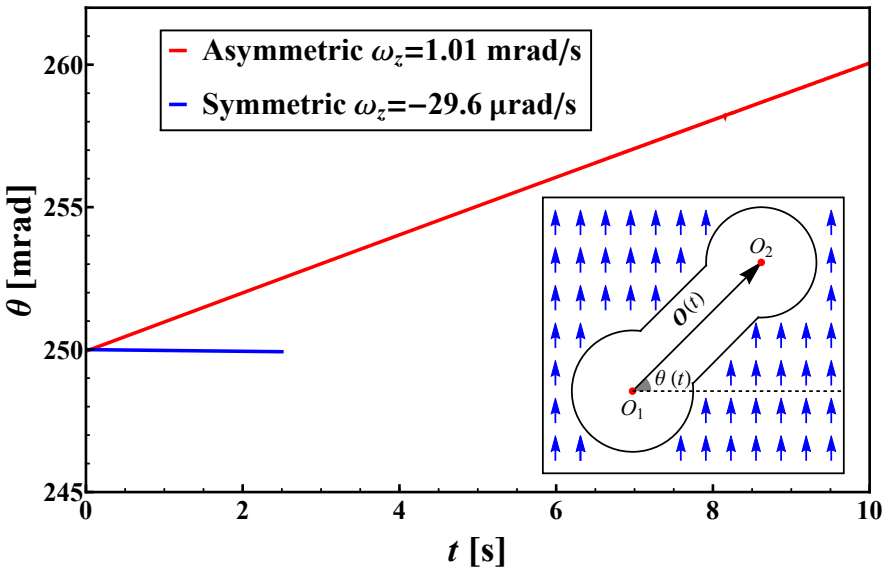


Figure 7: Tilt angle time evolution $\theta\,(t)$ for a symmetric and an asymmetric dumbbell. The latter exhibits positive rotation, which orients the smaller disc upstream. The symmetric dumbbell's rotation can be attributed to mesh defects. The inset shows how θ is defined.

fluid particle mass, number density and rotation angle are set to $m_\mathrm{f} = 1.00$, $\gamma = 5.03$ and $\alpha = \pi/2$, respectively. A Verlet-integration timestep and a collision timestep are choosen so $t_\mathrm{MD} = t_\mathrm{coll} = 4 \times 10^{-2}\, t_0$, therefore the mean free path is $\lambda = 4 \times 10^{-2}\, a$. This value of λ should yield a kinematic viscosity $\nu = 1.135\, a^2/t_0$. Setting $\mathrm{Re} = 0.1$, we calculate $g_y = 24 \times 10^{-6}\, a/t_0^2$. We run the simulation for 1.25×10^6 timesteps - the fluid is thermalised during the first 2.5×10^5, and the fluid particle coordinates and velocities are sampled every 1000 timesteps till the end of the simulation. A velocity profile is obtained by binning the particle velocities with respect to their z-coordinate and averaging within every bin. Since SRD is a stochastic method, we are supposed to run several simulations with different random seeds and average over that, as well, but given the size of the system and our sole interest in the steady-state profile, we simply average over time.

Due to limitations in the current implementation of SRD in LAMMPS we were only able to simulate a pure fluid between parallel plates or spheres immersed in an SRD fluid. The main goal here is to determine if SRD can outperform LBM. If this is indeed possible a series of improvements we intend to implement will be discussed in the next chapter.

Below we compare the velocity profile obtained for the system under consideration (Fig. 8) to the analytical one corresponding to the input parameters (red curve). The coincidence between theory and simulation results is quite good. The velocity profile obtained from the simulation is identical to the theoretical one safe for some minor thermal fluctuations inherent to the method. It is evident that LAMMPS' implementation of SRD recovers the correct behaviour of the system and does not suffer from drawbacks we encountered in LBM. This gives us encourages us to further develop the algorithm and make it capable of simulating dumbbells in a flow.

Finally, we compare the efficiency of the two methods. With LBM we managed to drive the dumbbell forward by roughly $L_y/2$ and the calculation took 5.5×10^5 s. We also ran a pure SRD fluid for $1.25 \times 10^6\, t_0$, which corresponds to an average displacement of most fluid particles by $L_y/4$. The total wall time of the computation in this case was 6.33×10^4 s. Although we cannot directly compare the two methods' performance (LBM is not well-optimized, while SRD does not support non-spherical colloids for now), we can speculate that both methods are capable of simulating the experimental system within a reasonable timeframe, provided some additions to both algorithms are made, a topic

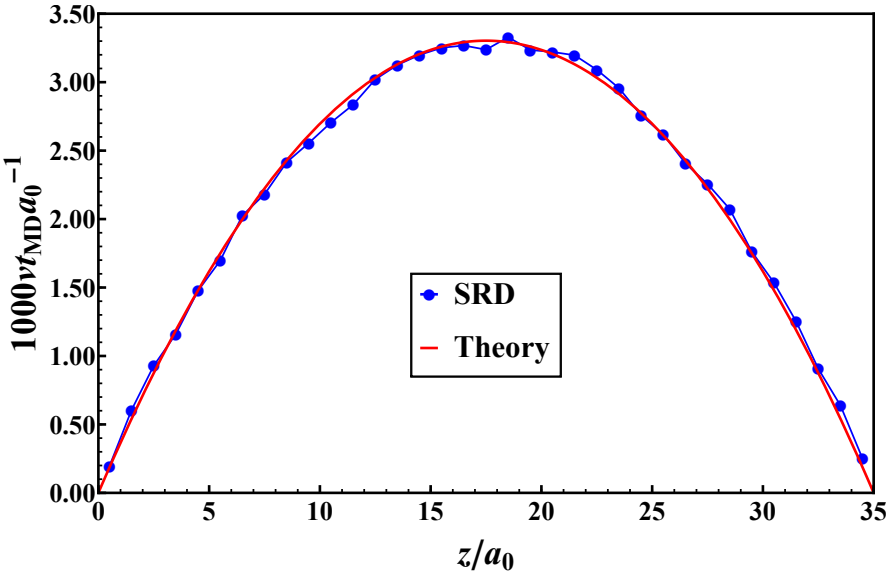


Figure 8: Poiseuille profile for an SRD fluid. The simulation result is almost identical to the analytical parabola. The observed thermal fluctuations are inherent to the algorithm and can be mitigated by seed averaging.

we will discuss in the next chapter.

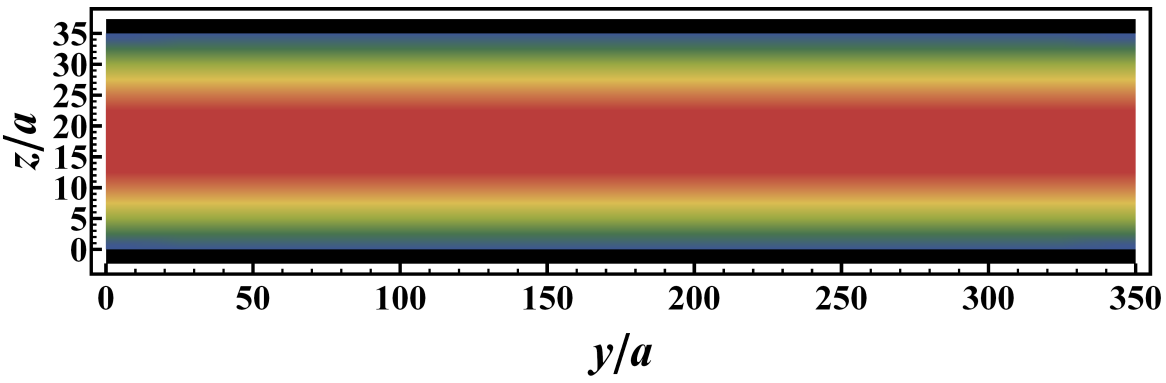
V. Discussion & Outlook

As we have already seen in the previous chapter the SRD and LBM algorithms in LAMMPS are not capable of recovering the oscillatory motion of a dumbbell confined to Q2D geometry, at least not in their current implementation. While LBM supports non-spherical particles, it is quite inefficient. In contrast, the SRD algorithm can be run within a reasonable time but only supports spherical colloids. In this chapter we discuss the possible code changes, which will enable us to simulate a dumbbell in Q2D without requiring expensive computational power and/or excruciatingly long runtimes.

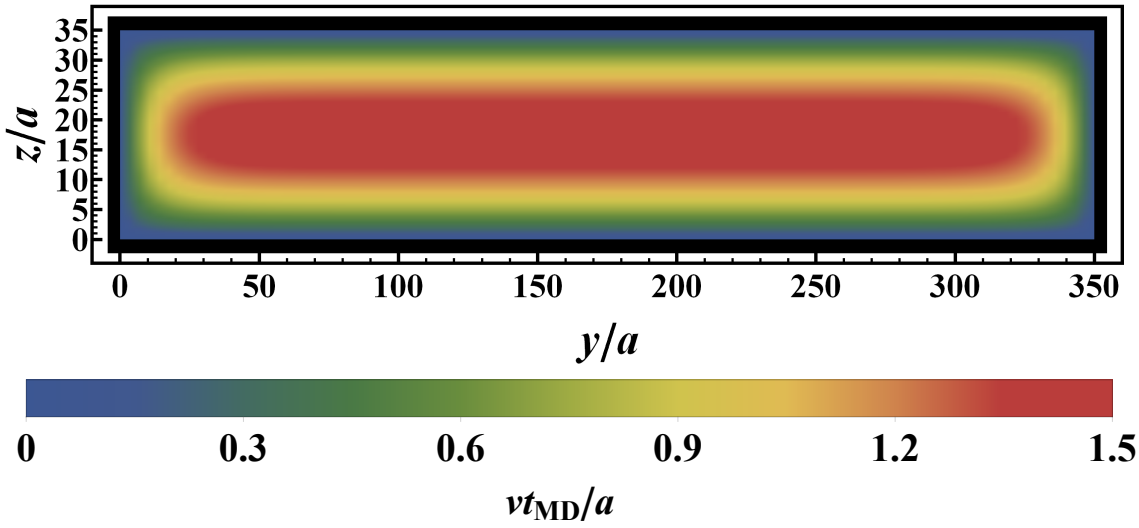
Second set of walls

The most important feature that needs to be added to both implementation is a second set of walls. While this will slow down the SRD algorithm it may reduce the number of timesteps needed to observe rotational motion in both SRD and LBM. The reason for that lies in the reduced symmetry of the flow when additional walls are added. When the fluid is confined between two plates only, the velocity profile is constant in the vorticity direction (Fig. 9a). An additional set of walls in that direction will result in two perpendicular superimposed Poiseuille profiles (Fig. 9b). In this case even a slight asymmetry in the dumbbell position or orientation with respect to the flow will result in rapid motion towards the wall. Furthermore, one will be able to observe the (damped) oscillatory movement of the dumbbell center of mass only when walls are present.

Implementing additional plates in LBM is trivial [17] and should slow down the algorithm only negligibly. This is not the case with SRD – proper coupling of the fluid to the walls requires the generation of positions $\boldsymbol{r}_{\mathrm{VP},i}$ and velocities $\boldsymbol{v}_{\mathrm{VP},i}$ for the virtual particles in the vorticity walls. Given the thickness of the system, these additional walls will not be large and thus not many $\boldsymbol{r}_{\mathrm{VP},i}$ and $\boldsymbol{v}_{\mathrm{VP},i}$ need to be generated. A major hindrance here would be the computation of the average velocity for



(a)



(b)

Figure 9: Fluid velocity field in the plane normal to the flow direction. (a) Plane Poiseuille flow – the system is periodic in the flow and vorticity directions. No symmetry breaking in the y-direction. (b) Rectangular duct flow – any assymetry in the particle position and/or orientation causes rapid lateral drift.

the given position of the virtual particle. Although an analytic expression for the steady state flow with two sets of walls is known [14], its implementation is computationally demanding. Nevertheless both editions hold promise to recover the motion of the colloid, known from experiments.

Fluid-colloid coupling in SRD

A major drawback of the current LAMMPS SRD algorithm is the lack of support for non-spherical particles immersed in the fluid. In some preliminary tests we have tried constructing a dumbbell from spheres held together by a rigid potential but the simulation proved unstable, fluid particles remained stuck in the spheres and eventually the simulation terminated. It is evident that the colloid-fluid coupling needs to be revised and improved.

Recently Poblete *et al.*[34] have suggested representing the colloid in a fashion similar to the one in LBM – a set of mesh point particles placed on its surface. In their treatment of the problem, momentum transfer between the fluid and the immersed body is carried out during the collision stage,

when the mesh particles take part in the collision as if they are heavier fluid particles. Straightforward as this solution may seem, it has one major drawback: it requires a complicated collision algorithm, which differentiates between fluid and colloid mesh particles. Nevertheless, the reported diffusion coefficients for spheres and the achieved no-slip boundaries make this work a natural starting point for the incorporation of non-spherical colloids in the LAMMPS SRD algorithm.

VI. Conclusion

Understanding the forces driving particle self-organization in confined low Reynolds number flows requires in-depth analysis of the flow disturbances induced by these colloids. Experimentally, this information is not easily accessible – one needs to use fluorescent tracer particles and track their motion. Apart from being challenging itself, these tracers may also change the dynamics in the system under consideration.

In this work we propose an alternative – we use two coarse-grained simulation methods which recover hydrodynamic interactions to obtain flow velocity fields around a dumbbell-shaped particle. Both Stochastic Rotational Dynamics and the Lattice Boltzmann Method are used as implemented in the multi-purpose simulation package LAMMPS. Though promising and well-optimized, the SRD-algorithm is incapable of immersing a non-spherical particle in the fluid. In contrast, LBM supports dumbbell-shaped particles, but is not well-optimized.

Despite these setbacks, we manage to obtain the velocity disturbance field a dumbbell introduces and it closely matches the dipolar disturbance theory [7] predicts. We also recover the lateral drift and rotation of the dumbbell, but due to computational limitations we are unable to draw any quantitative conclusions. The current work only presents preliminary results and seemingly poses more questions than it answers, but it also serves as a starting point for future investigation on the topic.

References

[1] Y. Xia, B. Gates and Z.-Y. Li, *Advanced Materials*, 2001, **13**, 409–413.

[2] Y. Lu, Y. Yin and Y. Xia, *Advanced Materials*, 2001, **13**, 415–420.

[3] D. Dendukuri, S. S. Gu, D. C. Pregibon, T. A. Hatton and P. S. Doyle, *Lab on a chip*, 2007, **7**, 818–828.

[4] D. Dendukuri, P. Panda, R. Haghgooie, J. M. Kim, T. A. Hatton and P. S. Doyle, *Macromolecules*, 2008, **41**, 8547–8556.

[5] D. Dendukuri and P. S. Doyle, *Advanced Materials*, 2009, **21**, 4071–4086.

[6] W. E. Uspal and P. S. Doyle, *Soft Matter*, 2012, **8**, 10676.

[7] W. E. Uspal and P. S. Doyle, *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 2012, **85**, year.

[8] W. E. Uspal and P. S. Doyle, *Soft Matter*, 2014, **10**, 5177–5191.

[9] A. Malevanets and R. Kapral, *The Journal of Chemical Physics*, 1999, **110**, 8605.

[10] G. R. McNamara and G. Zanetti, *Phys. Rev. Lett.*, 1988, **61**, 2332–2335.

[11] S. Plimpton, *Journal of Computational Physics*, 1995, **117**, 1–19.

[12] M. K. Petersen, J. B. Lechman, S. J. Plimpton, G. S. Grest, P. J. in âĂŹt Veld and P. R. Schunk, *The Journal of Chemical Physics*, 2010, **132**,.

[13] F. Mackay, S. Ollila and C. Denniston, *Computer Physics Communications*, 2013, **184**, 2021 – 2031.

[14] H. Bruus, *Microscale Acoustofluidics*, 2015, 1–28.

[15] J. T. Padding and A. A. Louis, *Phys. Rev. E*, 2006, **74**, 031402.

[16] G. Gompper, T. Ihle, D. M. Kroll and R. G. Winkler, *Advances in Polymer Science*, 2008, 1–34.

[17] S. Chen and G. D. Doolen, *Annu. Rev. Fluid Mech.*, 1998, **30**, 329–364.

[18] E. Allahyarov and G. Gompper, *Phys. Rev. E*, 2002, **66**, 036702.

[19] S. Wolfram, *J. Stat. Phys.*, 1986, **45**, 471–526.

[20] P. L. Bhatnagar, E. P. Gross and M. Krook, *Phys. Rev.*, 1954, **94**, 511–525.

[21] W. E. Uspal, H. Burak Eral and P. S. Doyle, *Nature communications*, 2013, **4**, 2666.

[22] M.-C. Lai and C. S. Peskin, *Journal of Computational Physics*, 2000, **160**, 705–719.

[23] T. Ihle and D. M. Kroll, *Phys. Rev. E*, 2003, **67**, 066705.

[24] C. Huang, A. Chatterji, G. Sutmann, G. Gompper and R. Winkler, *Journal of Computational Physics*, 2010, **229**, 168 – 177.

[25] J. K. Whitmer and E. Luijten, *Journal of Physics: Condensed Matter*, 2010, **22**, 104106.

[26] D. S. Bolintineanu, J. B. Lechman, S. J. Plimpton and G. S. Grest, *Phys. Rev. E*, 2012, **86**, 066703.

[27] M. Ripoll, K. Mussawisade, R. G. Winkler and G. Gompper, *Phys. Rev. E*, 2005, **72**, 016701.

[28] E. Bianchi, A. Z. Panagiotopoulos and A. Nikoubashman, *Soft Matter*, 2015, **11**, 3767–3771.

[29] A. Nikoubashman, C. N. Likos and G. Kahl, *Soft Matter*, 2013, **9**, 2603.

[30] A. Nikoubashman, *Soft Matter Advance Article*, 2017, –.

[31] N. Kikuchi, C. M. Pooley, J. F. Ryder and J. M. Yeomans, *The Journal of Chemical Physics*, 2003, **119**, year.

[32] C. Geuzaine and J.-F. Remacle, *International Journal for Numerical Methods in Engineering*, 2009, **79**, 1309–1331.

[33] P. R. Rinaldi, E. A. Dari, M. J. Vénere and A. Clausse, *Simulation Modelling Practice and Theory*, 2012, **25**, 163–171.

[34] S. Poblete, A. Wysocki, G. Gompper and R. G. Winkler, *Phys. Rev. E*, 2014, **90**, 033314.