Mesoscopic Modelling and Simulation of Soft Matter

Ulf D. Schiller [a], Timm Krüger [b] and Oliver Henrich *[c]

Abstract

The deformability of soft condensed matter often requires modelling of hydrodynamical aspects to gain quantitative understanding. This, however, requires specialised methods that can resolve the multiscale nature of soft matter systems. We review a number of the most popular simulation methods that have emerged, such as Langevin dynamics, dissipative particle dynamics, multi-particle collision dynamics, sometimes also referred to as stochastic rotation dynamics, and the lattice-Boltzmann method. We conclude this review with a short glance at current compute architectures for high-performance computing and community codes for soft matter simulation.

1 Introduction

Soft condensed matter [1,2] has an ubiquitous presence in our world and we encounter it in our everyday lives. Many complex fluids like polymer solutions and colloidal suspensions, liquid crystals, foams, gels, granular materials and biological materials belong to this category. A feature that distinguishes soft matter from more conventional condensed matter is that it shows a remarkable propensity to self-organise and form more complex, multiscale structures which exist on intermediate mesoscopic time and length scales much larger than the atomistic length scale, but also much smaller than the macroscopic lab scale. This characteristic nature of soft matter entails a typical separation of time and length scales and becomes important for solvent-mediated interactions e.g. between suspended nanoparticles or in systems with internal degrees of freedom like fluctuating membranes, polymer chains or vesicles. Thus, it can be quite challenging to find a consistent physical description that covers all relevant aspects of the problem.

Nonlinear coupling mechanisms between different components of the physical model, such as the order-flow coupling in liquid crystals, occur frequently in soft matter. Sometimes they prevent accurate analytical solutions and make simulations more favourable, or even outright indispensable. For a quantitative understanding of the response and dynamical behaviour it is crucial to find a suitable coarse-grained descriptions that manages to express a large number of degrees of freedom through a much smaller number of effective degrees of freedom whilst retaining the correct overall physical behaviour, therefore allowing to bridge time and length scales. An example is flowing soft matter. Quite generally speaking, flow is of particular importance due to the deformability of soft matter. Usually it can be assumed the flow is incompressible and characterised by low Reynolds numbers. Hydrodynamic interactions, however, lead to long-ranged, collective interactions that are notoriously difficult to treat with analytical models.

These examples highlight only a few complications on the way to model soft matter. They have led to the formulation of specialised simulation methods which are the focus of this tutorial review. Due to space limitations we can cover here unfortunately only the most common and versatile methods. In the following article, we will present a short overview of these relatively recent developments and glance also at some typical applications of these simulation methods and make suggestions for further reading.

1.1 Mesoscopic Modelling: Particle-based vs. Lattice Models

The mesoscopic methods discussed in this review are essentially alternative ways to model the dynamic correlations between solute particles that are mediated by momentum transport in a solvent medium. The momentum transport in the solvent is in principle described by the Navier-Stokes equation, however, the dynamics can also be affected by thermal fluctuations and specific molecular-level interactions. Mesoscopic methods are based on "coarse-graining" the microscopic details, i.e., including only the essential details of the interactions thus greatly reducing the degrees of freedom of the system. For instance, the hydrodynamic interactions between solute particles are amenable to a Langevin description

$$\dot{\mathbf{r}}_i = \sum_j \frac{\mathsf{D}_{ij}}{k_B T}\mathbf{F}_j + \Delta\mathbf{r}_i, \tag{1}$$

where $\mathbf{F}_j$ is the effective conservative force acting on particle j, D_{ij} is the diffusion tensor, and $\Delta\mathbf{r}_i$ are stochastic displacements that represent thermal fluctuations and satisfy the fluctuation dissipation relation

$$\langle \Delta\mathbf{r}_i \rangle = 0, \tag{2}$$

$$\left\langle \Delta\mathbf{r}_i(t)\Delta\mathbf{r}_j(t') \right\rangle = 2\mathsf{D}_{ij}\delta(t-t'). \tag{3}$$

In soft matter, diffusion of the solutes is much slower than diffusion of momentum leading to Schmidt numbers $Sc = \nu/D$ on the order of 10^6 for micron size particles. Any mesoscopic method should therefore guarantee $Sc \gg 1$. The diffusion tensor depends on the position of all particles in the system, and a simplified,

[a] Department of Materials Science and Engineering, Clemson University, 161 Sirrine Hall, Clemson, SC 29634, USA
[b] School of Engineering, University of Edinburgh, Edinburgh EH9 3FB, Scotland, UK
[c] Department of Physics, SUPA, University of Strathclyde, Glasgow G4 0NG, Scotland, UK, E-mail: oliver.henrich@strath.ac.uk

pair-wise additive form is given by the Rotne-Prager tensor[3]

$$\mathsf{D}_{ij} = D_0 \left\{ \left(1 - \frac{9\, r_{ij}}{32\, a}\right) \mathbf{I} + \frac{3}{32} \frac{\mathbf{r}_{ij} \otimes \mathbf{r}_{ij}}{a\, r_{ij}} \right\}, \qquad r_{ij} < 2a$$

$$\mathsf{D}_{ij} = D_0 \, \frac{3\, a}{4\, r_{ij}} \left\{ \mathbf{I} + \frac{\mathbf{r}_{ij} \otimes \mathbf{r}_{ij}}{r_{ij}^2} + \frac{2\, a^2}{3\, r_{ij}^2} \left(\mathbf{I} - 3 \frac{\mathbf{r}_{ij} \otimes \mathbf{r}_{ij}}{r_{ij}^2} \right) \right\}, \qquad r_{ij} \geq 2a$$

$$\mathsf{D}_{ii} = D_0 \, \mathbf{I} \tag{4}$$

where $\mathbf{I}$ is the identity matrix and a is the radius of the suspended particles and $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$. The equation of motion (1) can in principle be integrated numerically, and the most straightforward approach is known as Brownian dynamics[4]. The time evolution of the system can accordingly be written

$$\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + \frac{\mathsf{D}_{ij}}{k_B T} h \mathbf{F}_j + \sqrt{2h}\, \mathsf{B}_{ij} \cdot \mathbf{W}_j, \tag{5}$$

where $\mathbf{W}_j$ are random vectors representing a discretised Wiener process such that $\langle \mathbf{W}_i \rangle = 0$ and $\langle \mathbf{W}_i \otimes \mathbf{W}_j \rangle = \mathbf{I}\delta_{ij}$. The tensor B_{ij} is related to the diffusion tensor through $\mathsf{D}_{ij} = \mathsf{B}_{ik} \cdot \mathsf{B}_{jk}^T$ and can be represented by a Cholesky decomposition into an upper triangular matrix C_{ij} such that $\mathsf{D}_{ij} = \mathsf{C}_{ik} \cdot \mathsf{C}_{jk}^T$. The use of the Cholesky factorisation for BD was proposed by Ermak and McCammon[5], however, the algorithm scales as $O(N^3)$ with the particle number N and is infeasible already for a few hundred particles. Fixman[6] introduced a more efficient procedure using a Chebyshev polynomial approximation

$$\mathsf{B}_{ij} = \sum_{l=0}^{L} a_l C_l\left(\mathsf{E}\right) - \frac{a_0}{2} C_0\left(\mathsf{E}\right), \tag{6}$$

where $\mathsf{E} = (\mathsf{D}_{ij} - h_+)/h_-$, $h_+ = (\lambda_{\max} + \lambda_{\min})/2$ and $h_- = (\lambda_{\max} - \lambda_{\min})/2$, and $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalue of the diffusion tensor, respectively. The Chebyshev polynomials $C_l(\mathsf{E})$ are given by the recursion relation

$$C_0(\mathsf{E}) = \mathbf{I} \tag{7}$$

$$C_1(\mathsf{E}) = \mathsf{E} \tag{8}$$

$$C_{l+1}(\mathsf{E}) = 2\mathsf{E} \cdot C_l(\mathsf{E}) - C_{l-1}(\mathsf{E}). \tag{9}$$

The coefficients a_l are obtained from the Chebyshev series expansion of the square root function $f(e) = \sqrt{h_+ + h_- e}$

$$a_l = \frac{2}{L+1} \sum_{k=0}^{L} \cos\left(\frac{\pi l (k + \frac{1}{2})}{L+1} \right) f \left[\cos\left(\frac{\pi (k + \frac{1}{2})}{L+1} \right) \right]^{\frac{1}{2}}. \tag{10}$$

Instead of solving for B_{ij} directly, Fixman's algorithm uses the polynomial expansion is used to calculate the stochastic displacement

$$\mathsf{B}_{ij} \mathbf{W}_j = \sum_{l=0}^{L} a_l \mathbf{x}_l - \frac{a_0}{2} \mathbf{x}_0, \tag{11}$$

where $\mathbf{x}_l = C_l((\mathsf{D}_{ij} - h_+)/h_-) \cdot \mathbf{W}_j$. The Chebyshev polynomial approximation of $\mathsf{B}_{ij} \cdot \mathbf{W}_j$ for a given accuracy scales roughly as $O(N^{2.25})$ which can be reduced to $O(N^2)$ by truncated Chebyshev expansions. For further details and a comparison of different implementations we refer the reader to Ref. 7. The Chebyshev expansion may be sufficient to treat on the order of 10^3 particles, however, in practice the runtime behaviour depends strongly on the desired accuracy and the details of the underlying physics. For polymer chains, Schmidt et al.[7] have found that the performance of Chebyshev-based procedures is significantly affected by overlap of the chains. An advantage of BD methods is that they do not rely on an underlying simulation box and thus do not exhibit finite box size effects. Hence, for single chains, BD methods may be superior to explicit solvent methods due to the L^3 scaling of the fluid degrees of freedom. However, for semi-dilute polymer solutions, Pham et al.[8] have estimated that the scaling ratio between BD and explicit solvent (e.g. LB) methods will tip in favour of the latter. Larger BD systems thus require even faster algorithms such as Ewald-like methods using fast Fourier transformations[9]. These methods scale as $O(N^{1+x} \log N)$ where x is typically substantially smaller than unity. However, these methods require the study of confined systems and cannot easily be generalised to arbitrary boundary conditions. Moreover, the underlying Langevin description of BD neglects the retardation effect associated with the finite speed of momentum propagation in the solvent.

In this review, we therefore focus on mesoscopic methods that maintain a coarse-grained description of the solvent with explicit momentum transport. One can distinguish two broad classes of mesoscopic methods, namely particle-based and lattice models. Particle based-methods, such as dissipative particle dynamics (DPD) and multi-particle collision dynamics (MPC), represent the solvent by a system of interacting particles. The "coarse-graining" of the molecular details is achieved by implementing the interactions through collective collisions that satisfy the local conservation laws. Particle-based methods maintain a continuous phase space and thermal fluctuations are inherently present. DPD is essentially a momentum-conserving version of the Langevin thermostat and its algorithm is closely related to molecular dynamics based on Newton's equations of motion. In contrast, MPC is not developed as a time-discrete scheme for integrating a continuous equation of motion, but is based on discrete streaming and collision steps similar to Bird's direct simulation Monte Carlo (DSMC) method[10]. Both DPD and MPC can be shown to satisfy an H-theorem[11–13]. In addition, MPC can easily be switched from a micro-canonical ensemble to a canonical ensemble by augmenting the collision rule with a thermostat[14,15].

Lattice models, such as finite-element models or the lattice Boltzmann method (LBM) represent the solvent by hydrodynamic fields on a discrete lattice. Thermal fluctuations can be reintroduced by means of stochastic collisions, if needed. In lattice models, the solvent viscosity (and other transport coefficients) are directly linked to simulation parameters and thus can be set without the need for calibration. Moreover, the LBM can be rigorously derived from kinetic theory which provides a systematic route to extending its applicability to, e.g., multiphase fluids or Knudsen flows[16,17].

Both particle-based and lattice mesoscopic methods define a discrete dynamics that can be shown to reproduce Navier-Stokes hydrodynamics asymptotically. In this sense, the parameters of the coarse-grained model represent constitutive relations on the macroscopic level that can be tuned to reproduce the transport properties of the real physical system, cf. section 1.2. Moreover, the methods allow implementation of boundary conditions and provide systematic means of coupling solute particles to the solvent medium. For these reasons, mesoscopic methods are perfectly suited to study complex phenomena in soft matter systems, both in and out of equilibrium.

1.2 Parameter Choice in Mesoscopic Simulations

A crucial step in any mesoscopic simulation is the mapping of simulation parameters to physical quantities. Whereas many authors resort to what is commonly called "simulation units", there is often confusion as to how these are set and sometimes the details provided are insufficient to reproduce the results. Mesoscopic methods provide some flexibility in choosing a parameter mapping in such a way that the physical properties of interest are correctly captured by means of *dimensionless numbers*. The hydrodynamic transport properties of fluids are characterised by dimensionless numbers including the Schmidt number Sc, the Mach number Ma, the Reynolds number Re, the Knudsen number Kn, and the Péclet number Pe:

$$Sc = \frac{\nu}{D_f}, \tag{12}$$

$$Ma = \frac{u}{c_s}, \tag{13}$$

$$Re = \frac{uL}{\nu} = \frac{c_s L}{\nu} Ma, \tag{14}$$

$$Kn = \frac{\lambda}{L} \propto \frac{Ma}{Re}, \tag{15}$$

$$Pe = \frac{uL}{D_s} = \frac{\nu}{D_s} Re = \frac{c_s L}{D_s} Ma. \tag{16}$$

Here, λ is the mean free path, D_f is the self-diffusion coefficient of the solvent, and D_s is the diffusion coefficient of the solute particles. The value of D_s typically depends on the details of the coupling to the fluid for the respective method. In particle-based methods, a large Schmidt number can only be obtained if the mean free path is small, which usually requires a small time step h. This is a common characteristic of coarse-grained particle-based simulations that use a reduced number of particles which leads to a larger mean free path. In practice it is typically sufficient to ensure that the momentum transport is fast enough to generate liquid-like behaviour, and simulations can still be carried out with Schmidt numbers on the order of $\mathcal{O}(1)$ to $\mathcal{O}(10)$. Similarly, it is feasible to simulate at a higher Mach number than in the real system, as long as density fluctuations remain sufficiently small (incompressible limit).

Arguably the most important dimensionless number characterising hydrodynamic flow is the Reynolds number that quantifies the ratio of inertial and viscous momentum transport. In simulations of soft matter and microflows, the Reynolds number is typically small such that nonlinear inertial effects do not play an important role. The flow velocity is usually a result of external fields, and hence the Reynolds number can be used to tune the parameters that determine the magnitude of the driving forces of the simulation in relation to the viscosity of the fluid.

In terms of standard dimensionless groups, the ratio of the Mach number and the Reynolds number is proportional to the Knudsen number that quantifies the importance of rarefaction effects that can occur in microflows and lead to deviations from continuum Navier-Stokes behaviour. Knudsen numbers beyond $Kn \gtrsim 0.1$ indicate that the flow is in the transition regime where non-continuum effects can become significant. The Péclet number quantifies the relative importance of convective and diffusive transport of solutes which is related to the importance of thermal fluctuations, i.e., for small Péclet numbers Brownian motion dominates hydrodynamic advection. In microflows, the ratio Pe/Re is typically very large and it may not be feasible to use a sufficiently large number of particles to reproduce the Péclet number. In such systems one can seek a compromise and simulate at a higher Reynolds number and/or lower Péclet number, as long as one stays in the correct hydrodynamic regime which should be carefully validated.

In order to determine the simulation parameters, it is also instructive to consider the time scales of interest[18]. The main hydrodynamic time scales are the acoustic time scale $\tau_{c_s} = L/c_s$, the viscous (kinematic) time scale $\tau_\nu = L^2/\nu$, the diffusive time scale $\tau_D = L^2/D_s$, and the Stokes time scale $\tau_S = L/u$. These time scales can be related to each other using dimensionless numbers

$$\tau_S = \frac{\tau_{D_s}}{Pe} = \frac{\tau_\nu}{Re} = \frac{\tau_{c_s}}{Ma}. \tag{17}$$

This equation makes it evident how the separation between the time scales depends on the dimensionless numbers. It may thus not always be possible (and desirable) to resolve all time scales in the same simulation. In particular, the separation between the viscous and the acoustic time scale is typically large, which can be exploited to simulate the system with a speed of sound that is smaller than the real physical one.

2 Dissipative Particle Dynamics

Dissipative particle dynamics (DPD) is a stochastic simulation method that was specifically designed for soft matter and complex fluids. It was first formulated by Hoogerbrugge and Koelman[19,20] and later refined by Español[21] and Groot and Warren[22].

2.1 Basic Algorithm

The underlying idea is akin to coarse-grained molecular dynamics with atoms agglomerated into larger entities or "beads" that interact via soft forces. These beads are subject to conservative forces as well as pairwise drag or friction forces and random forces. The force balance for bead i is slightly different than in Eq. 1 as it runs over particle pairs (i, j):

$$m_i \ddot{\mathbf{r}}_i = \mathbf{F}_i^{\mathrm{C}} + \sum_{j \neq i} \mathbf{F}_{ij}^{\mathrm{D}} + \sum_{j \neq i} \mathbf{F}_{ij}^{\mathrm{R}}. \tag{18}$$

The functional forms of the forces are also slightly different. For instance, the conservative force

$$\mathbf{F}_i^{\mathrm{C}}(\{\mathbf{r}\}) = -\frac{\partial V(\{\mathbf{r}\})}{\partial \mathbf{r}_i} \tag{19}$$

always contains a soft-core repulsion in addition to other interactions or external forces.

The drag force depends on the relative separation $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and velocity $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ of the beads,

$$\mathbf{F}_{ij}^{\mathrm{D}}(\mathbf{r}_{ij}, \mathbf{v}_{ij}) = -\gamma\,\omega(|\mathbf{r}_{ij}|)\frac{(\mathbf{r}_{ij}\cdot\mathbf{v}_{ij})\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}, \tag{20}$$

whereas the random forces are given by

$$\mathbf{F}_{ij}^{\mathrm{R}}(\mathbf{r}_{ij}) = \sqrt{2k_BT\gamma\,\omega(|\mathbf{r}_{ij}|)}\,\frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}\,\xi_{ij}. \tag{21}$$

The quantity ξ_{ij} is a Gaussian random number with zero mean and unit variance which is symmetric with respect to the particle indices, i.e., $\xi_{ij} = \xi_{ji}$. This is a requirement for total momentum conservation and in contrast to Brownian and Langevin dynamics where the stochastic noise on each particle is independent of all other particles. γ is a friction coefficient and $\omega(|\mathbf{r}_{ij}|)$ a decaying weighing function with a model-specific cutoff distance[23,24]. In order to fulfil a fluctuation-dissipation theorem in DPD, distance-dependent friction forces require distance-dependent random forces for each pair of particles. This requirement has been demonstrated at Gibbsian equilibrium by Español and Warren[25].

Various schemes have been proposed to perform the time integration of Eq. 18[26]. The simplest integrator that can be considered a bare minimum requirement is based on a modified velocity-Verlet algorithm:

$$\begin{aligned}
\mathbf{v}_i(t+h/2) &= \mathbf{v}_i(t) + \frac{\sqrt{h}}{2m_i}\sum_{j\neq i}\mathbf{F}_{ij}^{\mathrm{R}}(\mathbf{r}_{ij}(t)) \\
&+ \frac{h}{2m_i}\left(\mathbf{F}_i^{\mathrm{C}}(\{\mathbf{r}(t)\}) + \sum_{j\neq i}\mathbf{F}_{ij}^{\mathrm{D}}(\mathbf{r}_{ij}(t), \mathbf{v}_{ij}(t))\right), \qquad (22) \\
\mathbf{r}_i(t+h) &= \mathbf{r}_i(t) + h\,\mathbf{v}_i(t+h/2), \qquad (23) \\
\mathbf{v}_i^{\star}(t+h) &= \mathbf{v}_i(t+h/2) + \frac{\sqrt{h}}{2m_i}\sum_{j\neq i}\mathbf{F}_{ij}^{\mathrm{R}}(\mathbf{r}_{ij}(t+h)) \\
&+ \frac{h}{2m_i}\mathbf{F}_i^{\mathrm{C}}(\{\mathbf{r}(t+h)\}), \qquad (24) \\
\mathbf{v}_i(t+h) &= \mathbf{v}_i^{\star}(t+h) \\
&+ \frac{h}{2m_i}\sum_{j\neq i}\mathbf{F}_{ij}^{\mathrm{D}}(\mathbf{r}_{ij}(t+h), \mathbf{v}_{ij}(t+h/2)). \qquad (25)
\end{aligned}$$

Note the square root of the time step h which appears in Eq. 22 and Eq. 24 when the stochastic Wiener process is discretised. In DPD the velocities $\mathbf{v}_i$ at the end of the time step depend on the drag forces $\mathbf{F}_{ij}^{\mathrm{D}}$ which in turn depend on the relative velocities. Hence, rather than just taking the drag forces based on the intermediate velocities at the half time step, a number of flavours of the DPD algorithm solve Eq. 25 in a self-consistent manner. In its simplest form the drag forces are recalculated once using the velocities $\mathbf{v}_i(t+h)$ as obtained in Eq. 25 and are then used in a final update of the velocities at $t+h$. This improves the performance significantly.

The local and pairwise interactions in DPD fulfil Newton's third law, conserve momentum and angular momentum, guarantee Galilean invariance and yield hydrodynamic conservation laws on larger length scales owing to the particle-based nature of the algorithm. A version of DPD with energy conservation can also be formulated[27]. Together with a modified predictor-corrector algorithm for the integration[22] or a self-consistent velocity-Verlet algorithm[28], DPD permits using larger time steps than atomistic MD modelling.

2.2 Model Extensions

A focus of interest has been around the thermodynamic consistency of DPD and free energy functionals[28]. If different species are modelled, the correct compressibility and solubility of the components, specified by the repulsion parameters between the different species, has to be provided[22]. Otherwise there is considerable freedom in modelling the interactions. DPD can also be derived from a coarse-grained version of molecular dynamics[29,30]. It is even possible to establish a link between DPD and a formulation of smoothed particle hydrodynamics (SPH)[31]. The resulting "smoothed" formulation of DPD is thermodynamically consistent and allows arbitrary equations of state.

Because pairwise interactions have to be calculated, DPD is computationally relatively expensive and "slower" than other methods, such as multi-particle collision dynamics or the lattice-Boltzmann method. Another disadvantage of DPD is that momentum transport is tightly coupled to particle transport and the Schmidt number is typically very low. However, schemes for arbitrarily large Schmidt numbers have been presented[32].

Further details on DPD can be found in the reviews by Nielsen[33], Moeendarbary[23,24] and Liu[34].

2.3 Applications

The DPD method has been used to model different dynamic regimes in polymer melts[35]. Here, the challenge lies in finding a coherent description for the crossover from Rouse dynamics of freely moving chains undergoing hydrodynamic interaction at short chain lengths to reptational dynamics at long chain lengths, where the individual chains are entangled and feel the topological constraints formed by other chains in their vicinity. Because of the softness of the beads, standard DPD models are unable to prevent unphysical bond crossings and disentanglement of the polymer chains. It is therefore necessary to apply the right degree of coarse graining by choosing the correct bead size and adapt the bond stretching interactions between the beads in such a way that disentanglement does not occur.

Another application area are electrohydrodynamic effects such as electro-osmosis and electrophoresis[36]. The method was recently used to model slip boundary conditions close to hydrophobic surfaces. Interestingly, it was shown that both confinement and mobility of the surface charges have a dramatic effect on the hydrodynamic properties of the electric double layer and the electro-osmotic flow[37].

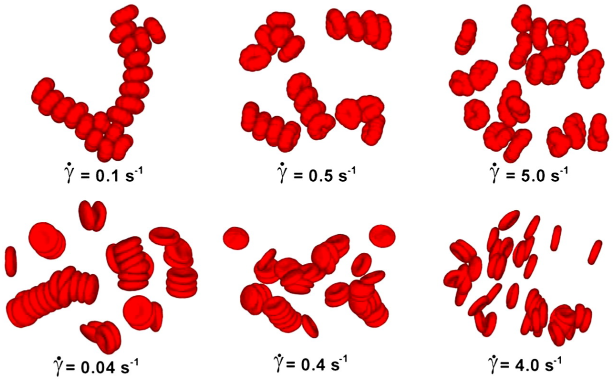
$\dot{\gamma}$ = 0.1 s^{-1}
$\dot{\gamma}$ = 0.5 s^{-1}
$\dot{\gamma}$ = 5.0 s^{-1}
$\dot{\gamma}$ = 0.04 s^{-1}
$\dot{\gamma}$ = 0.4 s^{-1}
$\dot{\gamma}$ = 4.0 s^{-1}

Fig. 1 Visualisation of aggregation of red blood cells. Simulated reversible rouleaux are formed by a low-dimensional model (upper) and multiscale model (lower). The left column corresponds to low shear rates, centre column to moderate share rates, and right column to high shear rates[38]. Copyright (2011) National Academy of Sciences.

DPD has been employed to simulate the dynamics and rheology of red blood cells (RBCs) suspended in blood plasma, cf. Fig. 1[38,39]. These cells are essentially biconcave viscoelastic shells (membranes) made of a lipid bilayer and a cytoskeleton, filled with a Newtonian haemoglobin solution. Numerically, all fluid and solid components are discretised as DPD particles: the external blood plasma with a viscosity of about $\eta_{\text{ex}} \approx 1.2$ mPa s, the internal haemoglobin solution ($\eta_{\text{in}} \approx 6$ mPa s), and the RBC membrane as a collection of DPD particles subject to additional viscoelastic forces. The membrane particles at locations $\{\mathbf{r}_i\}$ form a triangular mesh and have the potential

$$V(\{\mathbf{r}_i\}) = V_{\text{plane}} + V_{\text{bend}} + V_{\text{surf}} + V_{\text{vol}} \tag{26}$$

where V_{plane} is the in-plane shear contribution due to the RBC cytoskeleton, V_{bend} captures the bending resistance of the lipid bilayer, and V_{surf} and V_{vol} provide near conservation of the RBC surface area and volume which result from the incompressibility and impermeability of the RBC membrane. Additionally, in-plane viscous forces are taken into account that capture the viscosity of the lipid bilayer. Both external and internal liquid DPD particles are bounced back at either side of the membrane surface to ensure impermeability of the RBC membrane. The dissipative forces between membrane and liquid particles are specifically chosen to enforce the no-slip boundary condition.

3 Multi-Particle Collision Dynamics

Multi-particle collision dynamics (MPC), originally introduced by Malevanets and Kapral[11] as stochastic rotation dynamics (SRD) is a particle-based mesoscopic method that has become popular in the soft matter domain thanks to the flexibility in handling spatiotemporally varying forces. MPC is well suited to study complex phenomena in soft matter both in and out of equilibrium. In the following, we summarise the essential elements of MPC. A comprehensive review of MPC was published by Gompper et al.[40]

3.1 Algorithm

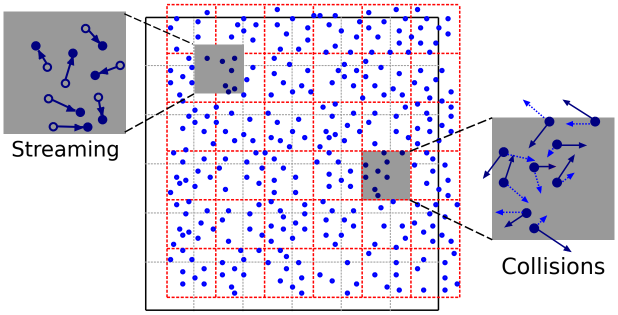

Fig. 2 The basic algorithm of multi-particle collision dynamics consists of successive streaming and collision steps. The cell grid is shifted randomly every time step to restore Galilean invariance[12,41].

In MPC, the fluid consists of idealised point-like particles, and the Navier-Stokes equation emerges from local mass and momentum conservation in the particle ensemble. The update of particle positions and momenta mimics the underlying kinetics and is defined in terms of successive streaming and collision steps, cf. Fig. 2. During the streaming step, the particles move ballistically

$$\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + h\mathbf{v}_i(t) \tag{27}$$

where h is the time interval between collisions. In the collision step, the particles are sorted into cubic collision cells of size a, and interactions occur only between particles within the same cell. The particles in each cell exchange momentum while the momentum of the collision cell is conserved

$$\mathbf{v}_i(t+h) = \mathbf{u}_C(t) + \mathbf{\Delta}_i(t) \tag{28}$$

where $\mathbf{u}_C$ is the velocity of the centre of mass of the cell, and $\mathbf{\Delta}_i$ is a collisional term that does not change $\mathbf{u}_C$. The centre of mass velocity in MPC plays a similar role as the equilibrium distribution in in the lattice-Boltzmann method (LBM).

The introduction of the collision cell introduces an artificial reference frame and breaks Galilean invariance. If the mean free path $\lambda = h\sqrt{k_BT/m}$ is smaller than the cell size a, repeated collisions between the same particles lead to correlations and the transport coefficients become dependent on imposed flow fields. In order to restore Galilean invariance, a random shift of the cell grid is performed before each collision step[12,41]. In practice, the shift can be performed by moving all particles by a random vector $\mathbf{s}$ with components uniformly distributed in $[-a/2, a/2[$ before the collision step and back after the collisions. The grid shift promotes momentum transfer between the cells and thus can lead to additional correlations in the transport coefficients[12,41]. Huang et al.[42] have carefully analysed the velocity correlations and the characteristic scales over which the correct hydrodynamic correlations and the long-time tail emerge.

The original multi-particle collision algorithm is referred to as stochastic rotation dynamics (SRD). It consists of a random rotation of the relative velocities $\mathbf{v}_{i,C} = \mathbf{v}_i - \mathbf{u}_C$ of the particles in a cell[11]

$$\mathbf{\Delta}_i^{\mathrm{SR}}(t) = \mathbf{v}_{i,C}^{\|}(t) + \mathbf{v}_{i,C}^{\perp}(t)\cos(\alpha) + \left(\mathbf{v}_{i,C}^{\perp}(t) \times \hat{\mathbf{a}}\right)\sin(\alpha) \qquad (29)$$

where $\hat{\mathbf{a}}$ is a randomly chosen axis, α the rotation angle, and $\mathbf{v}_{i,C}^{\|}$ and $\mathbf{v}_{i,C}^{\perp}$ denote the parallel and perpendicular components of $\mathbf{v}_{i,C}$ with respect to the random axis $\hat{\mathbf{a}}$. This collision rule is denoted as MPC-SR$-a$[43]. Instead of rotating the relative particle velocities, it is also possible to dampen the velocities with a Langevin thermostat (MPC-LD) or simply generate new relative velocities randomly. The latter is implemented by the algorithm

$$\mathbf{\Delta}_i^{\mathrm{AT}}(t) = \mathbf{v}_i^{\mathrm{ran}} - \sum_{j \in C} \frac{\mathbf{v}_j^{\mathrm{ran}}}{n_C} \qquad (30)$$

where $\mathbf{v}_i^{\mathrm{ran}}$ is a random velocity drawn from a Maxwell-Boltzmann distribution, and n_C is the number of particles in the collision cell. This collision rule is denoted as MPC-AT$-a$.[43] The random velocity serves as an Anderson-like thermostat to control the temperature, such that the simulation is effectively performed in a (local) canonical ensemble. For choices of alternative thermostats and their performance in equilibrium and non-equilibrium flows, we refer the reader to Refs. 14 and 15.

A drawback of both MPC-SR$-a$ and MPC-AT$-a$ algorithms is that they generate a non-symmetric stress tensor and hence do not conserve angular momentum. This can be mitigated by imposing angular momentum conservation as a constraint, which leads to an additional term in the collision rule

$$\mathbf{\Delta}_i^{+a}(t) = -\mathbf{r}_{i,C}(t) \times m\mathsf{I}^{-1} \sum_{j \in C} \left[\mathbf{r}_{j,C}(t) \times \left(\mathbf{v}_{j,C}(t) - \mathbf{\Delta}_j^{-a}(t)\right)\right] \qquad (31)$$

where m is the particle mass, I the moment of inertia tensor, and $\mathbf{r}_{i,C} = \mathbf{r}_i - \mathbf{r}_C$ the particle position relative to the centre of mass $\mathbf{r}_C$ of the cell. MPC-LD, MPC-AT, and MPC-SR with the $\mathbf{\Delta}^{+a}$ term do not conserve kinetic energy. For MPC-SR, a velocity rescaling can be applied to conserve energy, however, this breaks time-reversal symmetry and leads to deviations in the radial distribution function[43]. A collision rule that conserves both energy and angular momentum can be derived in two dimensions[44]. Various other collision rules have been proposed in the literature, and we refer the interested reader to the overview by Gompper et al.[40]

3.2 Transport Coefficients

Local hydrodynamic fields are defined for each MPC cell as

$$\rho(\mathbf{x}_C) = \frac{m}{a^3}\sum_{i \in C} 1, \quad \mathbf{j}(\mathbf{x}_C) = \frac{m}{a^3}\sum_{i \in C} \mathbf{v}_i, \quad e(\mathbf{x}_C) = \frac{m}{a^3}\sum_{i \in C} \frac{\mathbf{v}_i^2}{2}. \qquad (32)$$

These coarse-grained fields are the slow variables of the discrete-time dynamics that exhibit hydrodynamic behaviour on macroscopic time and length scales. The corresponding transport coefficients emerge from the micro-scale transport during streaming and collisions and hence contain both kinetic and collisional contributions. There are two possible routes to derive transport coefficients of the MPC fluid. The first uses a projection-operator formalism and relates the transport coefficients to equilibrium fluctuations of the hydrodynamic fields[45–47]. In the second approach, transport coefficients are determined through analysis of non-equilibrium steady-state situations. By virtue of the fluctuation-dissipation theorem, the linear response of the system to imposed gradients allows to calculate transport coefficients that are identical to the ones obtained from equilibrium fluctuations[43,48,49].

The latter approach leads to generic expressions for the diffusion coefficient,

$$D = \frac{k_B T h}{m}\left(\frac{1}{s} - \frac{1}{2}\right), \qquad (33)$$

and the kinetic and collisional contributions to the shear viscosity,

$$\nu^{\mathrm{kin}} = \frac{k_B T h}{m}\left(\frac{1}{c} - \frac{1}{2}\right), \qquad (34)$$

$$\nu^{\mathrm{col}} = \frac{g}{12}\frac{a^2}{h}, \qquad (35)$$

where the correlation factors s, c and g depend on the specific collision rule and are given in Table 1. Other transport coefficients can be derived along similar lines. In addition to mass and momentum transfer, the energy conserving versions of MPC are able to reproduce the heat transfer in the fluid. The thermal diffusivity and the thermal conductivity are given by[47,49]

$$D_T^{\mathrm{kin}} = \frac{k_B T h}{m}\left(\frac{1}{\theta} - \frac{1}{2}\right), \qquad (36)$$

$$D_T^{\mathrm{col}} = \frac{a^2}{3(d+2)h}\frac{1-\cos\alpha}{N}. \qquad (37)$$

A summary of the results is reported in Table 1. The similarity of the kinetic viscosity with the corresponding expression for the LBM should be noted. The latter can indeed be derived with the approach used in this section, which shows that the kinetic viscosity of both MPC and LBM is that of an ideal gas. The LBM does not have an analogue of the collisional viscosity, however, because the LBM collision process is entirely local and does not transfer momentum. This is an essential difference between particle-based and fully kinetic methods, and here MPC can be seen as a bridge between these two approaches.

3.3 Boundary Conditions and Suspended Objects

If the fluid is bounded by external walls or contains solid obstacles, no-slip boundary conditions at the solid surface are commonly imposed. A standard mesoscale procedure is to apply the bounce-back rule where the velocity $\mathbf{v}$ of an impinging particle is reversed to $-\mathbf{v}$. This procedure can be used if the boundaries coincide with the boundaries of the collision cells. However, due to the grid shift this is not always the case, and the boundaries generally intersect collision cells which can lead to an artificial slip velocity. This consequence of partially filled cells can be mitigated by filling the intersected cells with virtual particles such that the total number of particles matches the average of N particles per cell. The MPC collision step is performed using the mean velocity of all particles in the cell, where the velocities of the virtual particles are drawn from a Maxwell-Boltzmann distribution with zero mean velocity[50]. The virtual particles ensure

Table 1 Correlation factors for MPC collision algorithms. Note that the MPC+a expressions for s, c and g are approximations for large N. For small N, additional correction terms have to be taken into account[43]. The factor $f = \frac{N-1+\exp(-N)}{N}$ is obtained by averaging over a Poisson distribution for the cell occupation number n_C with $\langle n_C \rangle = N$. For angular momentum conserving collision rules, additional correlations are present which result in the more complicated expressions. It should be noted that in the expression for θ terms of $O(1/N^2)$ have been neglected[45].

correlation factor		MPC-SR	MPC-LD	MPC-AT$-a$	MPC-AT$+a$
s		$\frac{2}{d}(1-\cos\alpha)f$	$\frac{\gamma h/m}{1+\gamma h/2m}$	f	$1-\frac{d+1}{2N}$
c	$d=2$	$(1-\cos 2\alpha)f$	$\frac{2\gamma h/m}{(1+\gamma h/2m)^2}$	f	$\frac{d}{2N}+1\left(1-\frac{3d+2}{4N}\right)$
	$d=3$	$\frac{2}{5}(2-\cos\alpha-\cos 2\alpha)f$			
g		$\frac{2}{d}(1-\cos\alpha)f$	$\frac{\gamma h/m}{1+\gamma h/2m}$	f	$\frac{1}{2}\left(1-\frac{7}{5N}\right)$
θ		$\frac{2}{d}(1-\cos\alpha)$ $-\frac{4}{dN}(1-\cos\alpha)^2\left(\frac{7-d}{5}-\frac{1}{4}\csc^2\frac{\alpha}{2}\right)$			

that a no-slip boundary condition is obtained which has been verified numerically for Poiseuille flows and flows around circular and square cylinders[50]. For cases where the boundary location does not coincide with the average center of mass of the particles, artificial slip can be reduced by allowing fluctuations in the number particles number such that the number of particles is Poisson distributed[14,15,51].

A similar approach can be used to impose boundary conditions on suspended objects such as colloidal particles. A fluid particle that collides with the colloid acquires a new velocity where the normal and tangential components are drawn from Maxwellian-like distributions[52]. In dense suspensions, there may be multiple collisions with more than one colloid (or a wall) during a time step, and repeated scattering is necessary to avoid an artificial depletion interaction between the colloids. The thermal wall approach is effective for suspended objects that are much larger than the mean free path λ.

For suspended objects with internal degrees of freedom, such as polymers or cells, a different coupling approach can be used. The particles are typically updated by a molecular dynamics scheme that takes into account intra- and inter-molecular forces. For the coupling to the fluid, the particles or monomers are considered point-like and interact with the fluid particles by means of the MPC collisions. The momentum exchange between the fluid particles and the monomers can be included as force in the molecular dynamics update. In order to accurately capture the hydrodynamic interactions, the average number of monomers per collision cells should be smaller than 1, and the monomers should be neutrally buoyant, i.e, the monomer mass m_m on the order of solvent mass per cell $n_c m$. Typically, the size a of the MPC cells has to be on the order of the bond length of the polymer. The MPC point-particle coupling has been used extensively to simulate polymeric and colloidal suspensions[40].

3.4 Non-Ideal Fluids, Binary Mixtures, Viscoelastic Fluids

In order to model dense gases and non-ideal liquids, excluded volume effects have to be incorporated. Generally, any non-ideal effect makes it necessary to consider non-local interactions. This can be achieved by dividing a collision cell into sub-cells with side length $a/2$ which exchange momentum during the collision process[53,54]. The equation of state of the non-ideal MPC fluid can be found from the mechanical definition of pressure. The non-ideal algorithm does not conserve phase-space volume because the collision probability depends on the difference of sub-cell velocities which can be realised by different states. However, careful consistency checks have not revealed a violation of detailed balance or other inconsistencies[54].

Repulsive interactions between different species of a binary mixture can be achieved in a similar manner as for excluded volume interaction between sub-cells. However, instead of exchanging momentum between the entire sub-cells, only collisions between particles of type A and B are taken into account. Additional MPC collisions at the cell level are incorporated for momentum exchange between particles of the same type A or B which allows to tune the overall viscosity[53–55]. The mixture model can be extended to amphiphilic and ternary fluids and allows to study the phase behaviour of complex fluids on large time scales. An alternative approach to binary mixtures is the introduction of a colour charge, similar to colour models in the LBM. The collision process takes into account the concentration of the species by exchanging momentum such that the colour-weighted momentum in each collision cell points in the direction of colour gradient. While it can be used in two and three dimensions, the colour model does not include thermal fluctuations of the order parameter. Nonetheless, it has been shown that his model leads to phase separation, satisfies the Laplace equation and can be extended to simulate ternary mixtures[56–59].

Viscoelastic behaviour can be modelled within MPC by using dumbbell springs instead of point particles. The MPC algorithm can still be performed in the usual manner, where in the streaming step the centre of mass of the dumbbells moves ballistically while the relative coordinates are updated according to the bond interaction. The collision step is applied to the end-points of the dumbbells and proceeds in the usual way for the various collision rules. An MPC fluid consisting of harmonic dumbbells can capture the orientation and elongation in shear flow and thus reproduces the viscoelastic behaviour of a Maxwell fluid[60]. However, due to the possible infinite bond extension, harmonic dumbbells do not reproduce non-equilibrium properties such as shear-thinning. If FENE dumbbells are employed instead, the proper behaviour corresponding to a dilute polymer solution is found[61]. Another alternative is to introduce a constraint of constant mean square bond length where the equilibrium value corresponds to Gaussian chains[62]. The dumbbell fluid exhibits shear thinning, where at large shear rates $\eta_b \sim Wi^{-2/3}$ with the Weissenberg number $Wi = \dot{\gamma} k_B T/(2DK_0)$. Moreover, due to hydrodynamic interactions, the fluid exhibits a non-vanishing second normal stress with the same shear rate dependence[62].

3.5 Applications

Like in other mesoscopic methods, MPC has been applied to a wide range of colloidal and polymeric systems[40,63,64]. The capabilities of MPC for simulating complex fluids are exemplary highlighted for the hydrodynamics of star polymers in shear flow[65–67]. Star polymers are modelled as linear polymer chains that are linked to a common monomer at the centre. A shear flow is induced in the MPC solvent by employing Lees-Edwards boundary conditions. The imposed flow field is strongly screened inside the star polymer, which was confirmed by comparing simulations with and without hydrodynamic interactions, respectively[66]. The results from MPC simulations were found to agree well with experimental measurements[67].

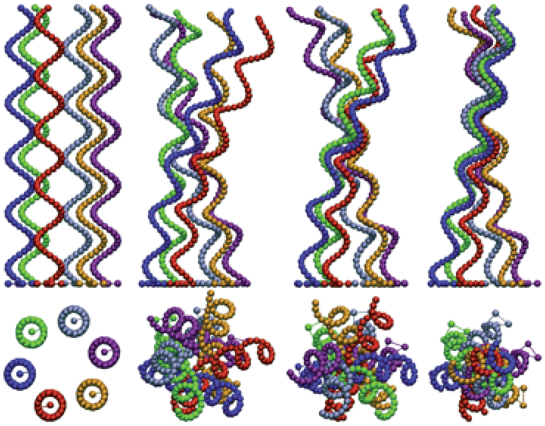

Fig. 3 Synchronisation and bundling of helical flagella are governed by hydrodynamic interactions and can be modelled using an MPC fluid. Reproduced from Ref. 68 with permission from the Royal Society of Chemistry.

MPC can also be applied to non-equilibrium flows where analytical theories are still limited due to the lack of applicable variational principles. One particular example of a driven dissipative system are microfluidic droplets in a Hele-Shaw geometry. A train of microfluidic droplets can be modelled within 2D-MPC using a frictional coupling on disc-like droplets[69,70]. The results confirm quantitatively that a far-field approximation of the dipolar hydrodynamic interactions remains valid even at high densities[69]. Moreover, a confinement-induced coupling of longitudinal and transverse oscillations was discovered where the longitudinal motion of the droplets is induced by the boundary conditions of the flow field[69,70]. In the presence of thermal fluctuations, the droplet oscillations exhibit instabilities that were also observed in simulations[70]. MPC thus paves the way to systematic investigation of the governing principles of non-equilibrium systems using well controlled model systems that are at the same time promising to develop novel methods in statistical physics.

Another area of high interest where timely contributions have been made using MPC is the hydrodynamics of swimmers[71,72]. Swimming behaviour can be produced by a number of mechanisms which differ in the characteristics of the flow field they produce and, consequently, the governing hydrodynamic interactions. For example, peritrichous bacteria use rotating helical flagella for self-propulsion, cf. Fig. 3. The mechanism of synchronisation and bundling of flagella has been studied using MPC[68,73,74]. The dependence of characteristic times for synchronisation and bundling on the number of flagella, separation, and motor torque was found to be governed by hydrodynamic interactions. The swimming properties of a flagellar model bacterium consisting of a spherocylindrical body have also been simulated using MPC[74]. These simulation models can be extended to study swimming near surfaces where confinement effects and non-equilibrium effects are expected to play an important role, similar to the microfluidic droplet systems described above.

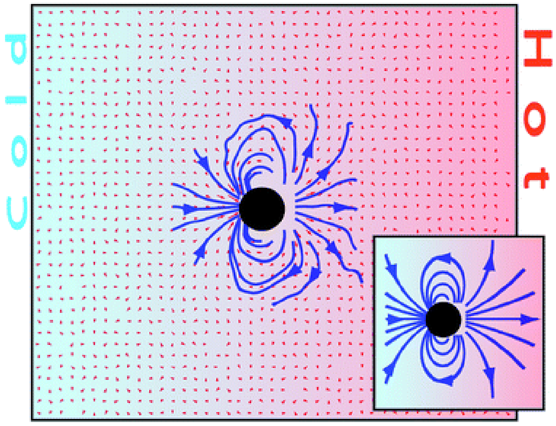

Fig. 4 Flow field around a freely drifting colloidal sphere in a temperature gradient. Reproduced from Ref. 75 with permission from the Royal Society of Chemistry.

MPC is uniquely suited to study inhomogeneous systems with temperature gradients. The investigation of thermodiffusion and thermophoresis using MPC was pioneered by Ripoll and coworkers[75–81]. The flow field induced by thermophoretic motion of a colloidal particle in an MPC fluid was found to be Stokeslet-like for a fixed particle and dipolar for a freely drifting particle, cf. Fig. 4[75]. The study of thermophoretic motion using MPC has inspired a number of applications, for instance, the induced flow field around a fixed colloidal particle gives rise to a net solvent flow which can be exploited as a microfluidic pump[75]. Another interesting application is the case where the temperature gradient is generated locally by a solute particle, e.g., by inhomogeneous light absorption or inhomogeneous chemical reactions. This idea was subsequently used to design a thermophoretic nanoswimmer consisting of two linked monomers and self-phoretic Janus particles[76,78]. Rotational motion due to thermophoresis was first demonstrated in MPC simulations of a micro-gear[79], and subsequently for a micro-turbine with anisotropic blades[80] and a catalytic micro-rotor driven by diffusiophoresis in a concentration gradient[81].

4 Lattice Boltzmann Method

The lattice Boltzmann Method (LBM)[82–86] is a lattice-based model unlike the particle-based methods described in the previous sections. Historically, it has not evolved from an adaptation of a flavour of molecular dynamics, but from lattice gas cellular automata[87]. LBM was originally introduced by McNamara and Zanetti[88] to mitigate the statistical noise that plagued the lattice gas automata. LBM is rooted in the kinetic theory of gases, and solves a simplified and discretised Boltzmann equation.

Characteristic for LBM is the full discretisation of time, configuration and velocity space[89]. The configuration and velocity discretisation are matched in such a way that the resulting algorithm becomes favourably simple and nearly local. These features lead to high numerical efficiency and parallelisability which are today extensively exploited throughout the soft matter community.

4.1 Basic Algorithm

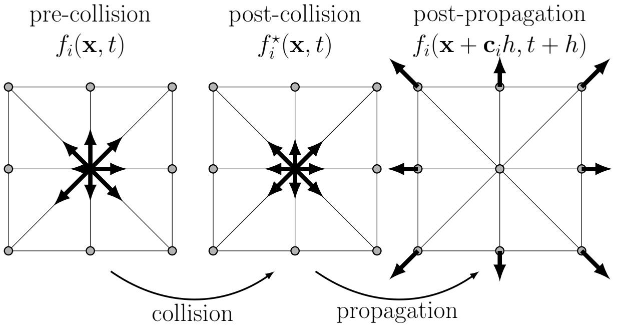

Fig. 5 The lattice Boltzmann equation, Eq. (38), can be understood as subsequent local collision and non-local propagation steps. Each arrow indicates the magnitude and direction of one of the nine distributions f_i on a 2D lattice during each of the stages of the algorithm. Lattice nodes are shown as grey circles.

The central quantity for LBM is the probability density function $f(\mathbf{x},\mathbf{c},t)$ which measures the probability of finding a particle with velocity $\mathbf{c}$ at position $\mathbf{x}$ at time t. Under the assumption of molecular chaos, f is a one-particle distribution function, i.e., the velocities of all colliding particles are uncorrelated and independent of their positions. Molecular chaos implies that the Boltzmann equation can be closed, i.e., the collision terms does not include the two-particle distribution function. This assumption is well justified for gases and fluids which are not too dilute (low Knudsen number). Then f characterises the dynamic state and obeys the Boltzmann equation. The basic algorithm of its simplified and discrete analogue, the lattice Boltzmann equation (LBE), is shown in Fig. 5 and can be summarised as

$$f_i(\mathbf{x}+\mathbf{c}_i h, t+h) = f_i^\star(\mathbf{x},t) = f_i(\mathbf{x},t) + \Omega_i(\mathbf{x},t) + S_i(\mathbf{x},t) \qquad (38)$$

with Ω_i and S_i explained below. The velocities $\mathbf{c}_i$ are a set of discrete lattice vectors that connect each lattice site to its nearest neighbours. Due to the discrete nature of the lattice vectors, the isotropy is generally broken. However, through an appropriate choice of the $\mathbf{c}_i$, isotropy can be restored to the required extent, and the resulting n-dimensional LBM schemes with m velocities are often denoted DnQm in the popular classification according to Qian[90].

The term Ω_i in Eq. (38) is the collision operator that describes the effective intermolecular interactions that modify the distribution function f_i related to each lattice direction. It can be written in the general form

$$\Omega_i = -\sum_j \Lambda_{ij}\left(f_j - f_j^{\mathrm{eq}}\right) = -w_i \sum_k b_k^{-1} e_{ki} \lambda_k \left(m_k - m_k^{\mathrm{eq}}\right) \qquad (39)$$

where the eigenvalues λ_k are relaxation coefficients and the moments m_k are obtained from the orthogonal transformation

$$m_k = \sum_i e_{ki} f_i, \qquad f_i = w_i \sum_k b_k^{-1} e_{ki} m_k \qquad (40)$$

and e_{ki} are orthogonal basis vectors with respect to the scalar product $\sum_i w_i e_{ki} e_{li} = b_k \delta_{kl}$. The basis vectors e_{ki}, weights w_i, and norms b_k depend on the specific lattice model in use[91,92]. This model is called multiple relaxation times (MRT). The local equilibrium values $m_k^{\mathrm{eq}} = \sum_i e_{ki} f_i^{\mathrm{eq}}$ are chosen such that the macroscopic equations are correctly recovered and are often based on a low-velocity expansion of the Maxwell-Boltzmann distribution:

$$f_i^{\mathrm{eq}}(\rho,\mathbf{u}) = w_i \rho \left(1 + \frac{\mathbf{u}\cdot\mathbf{c}_i}{c_s^2} + \frac{\mathbf{uu} : (\mathbf{c}_i\mathbf{c}_i - c_s^2 \mathsf{I})}{2c_s^4}\right). \qquad (41)$$

The simplest choice of the eigenvalues is the single relaxation time approximation $\lambda_k = \tau^{-1}$, known as the lattice-BGK collision model named after Bhatnagar, Gross and Krook (BGK), or single relaxation time (SRT). The relaxation time τ is related to the kinematic viscosity ν via $\nu = (\tau - h/2)c_s^2$ with the speed of sound c_s. The MRT model offers a larger number of free parameters than BGK; these parameters can be used to increase accuracy and stability of the algorithm. MRT also allows to have different shear and bulk viscosities. Other collision operators are the so-called two relaxation times (TRT)[93] or entropic LBM[94].

The source term S_i in Eq. (38) is related to the local volumetric force $\mathbf{F}$ through[95]

$$S_i = \sum_j \left(\delta_{ij} - \frac{1}{2}\Lambda_{ij}\right) w_j \left(\frac{\mathbf{F}\cdot\mathbf{c}_j}{c_s^2} + \frac{\mathbf{uF} : (\mathbf{c}_j\mathbf{c}_j - c_s^2 \mathsf{I})}{2c_s^4}\right). \qquad (42)$$

The density ρ, momentum density $\rho\mathbf{u}$, and momentum flux $\mathbf{\Pi}$ of the fluid are the zeroth, first, and second moments, respectively[86,92]:

$$\rho = \sum_i f_i, \quad \rho\mathbf{u} = \sum_i \mathbf{c}_i f_i + \frac{\mathbf{F}}{2}, \quad \mathbf{\Pi} = \frac{1}{2}\sum_i (f_i + f_i^\star)\mathbf{c}_i\mathbf{c}_i. \qquad (43)$$

The momentum flux $\mathbf{\Pi} = \mathbf{\Sigma} + \boldsymbol{\sigma}$ contains the Euler stress $\mathbf{\Sigma} = \rho\mathbf{uu} + p\mathbf{I}$ and the deviatoric stress tensor $\boldsymbol{\sigma} = \rho\nu[\nabla\mathbf{u} + (\nabla\mathbf{u})^\top]$. The link between the LBE (Eq. 38) and the macroscopic Navier-Stokes equation can be established through a Chapman-Enskog expansion[83,86,92].

A direct consequence of the discreteness of the underlying lattice is that Galilean invariance is broken in the LBM, which manifests itself as $O(u^2)$ errors in the viscosity. Hence the LBM is valid in the incompressible limit which corresponds to low Mach numbers. It is also possible to devise mitigating schemes[96,97]. They tend, however, to take away some of LBM's simplicity.

In the standard LBM algorithm, hydrodynamic flow occurs without thermal fluctuations. A scheme of fluctuating LBM can be devised which satisfies a fluctuation-dissipation theorem[98]. The thermalisation of the so-called ghost modes, the additional degrees of freedom that are not directly related to the ten hydrodynamic observables, lead to equipartition of the fluctuating energy on all length scales. This scheme has been recently extended from single phase fluids to binary mixtures[99].

4.2 Boundary Conditions

The LBM knows countless boundary conditions[86]. The reason for this plethora is that there are no unique links between the macroscopic Dirichlet and Neumann boundary conditions and the boundary conditions for the kinetic quantities f_i. Here, we will briefly mention only the most common boundary conditions for soft matter applications.

The bounce-back (BB) boundary condition[100] takes direct advantage of the kinetic nature of LBM. As such, there is no counterpart in conventional Navier-Stokes solvers, showing the advantage of LBM for applications with complex geometries. A post-collision population $f_i^\star$ initially moving from a fluid site $\mathbf{x}$ to a solid site $\mathbf{x}+\mathbf{c}_i h$ reverts its direction (denoted by $\mathbf{c}_{\bar{i}} = -\mathbf{c}_i$) during the BB procedure, only to reach its origin at the end of the propagation step:

$$f_{\bar{i}}(\mathbf{x}, t+h) = f_i^\star(\mathbf{x}, t) - 2w_i \rho \frac{\mathbf{c}_i \cdot \mathbf{u}_\text{b}}{c_\text{s}^2} \tag{44}$$

where the second term on the right-hand side is the momentum exchange due to the boundary moving with velocity $\mathbf{u}_\text{b}$. Eq. (44) replaces the LBE in Eq. (38) if the node at $\mathbf{x}+\mathbf{c}_i h$ is solid rather than fluid.

BB leads to a no-slip boundary condition at a plane half-way between the fluid and the solid nodes with second-order accuracy[101]. Despite its staircase-like discretisation that can lead to numerical artefacts, the BB method is stable, simple to implement and computationally efficient. It is often employed for colloidal particles and porous media[102,103]. More sophisticated boundary conditions exist that allow boundaries at other locations than half-way between lattice nodes, for example interpolated BB[104] or multireflection[105]. We refer to a comprehensive review[92] for further information on fluid-structure interactions for soft matter applications.

In order to model immersed soft objects, such as polymers[106] and biological cells[107], the immersed boundary method (IBM)[108] is commonly used. The basic idea of the IBM is to introduce a Lagrangian mesh that captures the shape of the immersed object and can move relatively to the Eulerian lattice of the fluid. The mesh and the lattice are coupled through velocity interpolation (Eulerian to Lagrangian) and force spreading (Lagrangian to Eulerian) steps. It is possible to equip the Lagrangian objects with suitable elastic properties, such as dilation, shear and bending resistance.

Several works have been published about the combination of thermal fluctuations and immersed objects, such as colloids or polymers. Ahlrichs and Dünweg[106] added thermal fluctuations to the particle phase and restored momentum conservation by including opposite fluid forces. Contrarily, Ollila et al.[109] claim that the addition of Langevin noise to the particle phase can be avoided, an idea that is closer to Einstein's original notion of Brownian noise. There does not seem to be a consensus under which conditions both approaches are equivalent or unphysical.

4.3 Multi-Phase and Multi-Component Flows

Multi-phase or multi-component flows are ubiquitous in soft matter systems and occur generally in mixtures of different species, heterogeneous systems like fluid-solid or fluid-gas mixtures or when interfaces are present. LBM offers a convenient route to describe the forces arising between the individual phases and components in a consistent way, and probably even more importantly, on the mesoscopic level and in complex geometries.

Several methods have emerged to date. The Shan-Chen (SC) model[110], for instance, is a phenomenological approach which is based on so-called pseudo-potentials that mimic the microscopic interactions between the constituents. For a fluid mixture with an arbitrary number of components, each component σ is subjected to the short-range interaction force

$$\mathbf{F}^\sigma(\mathbf{x}) = -\psi^\sigma(\mathbf{x}) \sum_{\sigma'} g_{\sigma\sigma'} \sum_i w_i \psi^{\sigma'}(\mathbf{x}+\mathbf{c}_i h)\mathbf{c}_i h \tag{45}$$

where $g_{\sigma\sigma'} = g_{\sigma'\sigma}$ is the interaction strength between components σ and σ'. Note that $\sigma = \sigma'$ is permitted, which allows for self-interaction and non-ideal fluids with phase change. The pseudo-potential $\psi^\sigma(\mathbf{x})$ is a (usually linear or exponential) function of the component density $\rho^\sigma(\mathbf{x})$ and has been introduced to increase the numerical stability of the method. Surface tension emerges from the component interaction; its value γ is a function of the interaction strength and the chosen pseudo-potential.

An alternative approach is based on free energy (FE) models[111,112]. In contrast to the SC method with its emergent surface tension properties, FE methods are based on a top-down approach, starting directly from the relevant target equation that describes the dynamics of the complex fluid. Therefore, opposed to the SC model, thermodynamic consistency is straightforwardly realised in FE models. This is because the latter use the free energy functional as input, which means thermodynamic coupling terms between order parameters can be directly specified. Hence, FE models can be systematically extended to incorporate additional physical effects. They also provide better control of the thermodynamic state, an important aspect close to interfaces and boundaries. For example, the Landau free energy density for a binary mixture of two immiscible liquids with identical density ρ for both liquids can be written as[113]

$$\psi = c_\text{s}^2 \rho \ln \rho + \frac{a}{2}\phi^2 + \frac{b}{4}\phi^4 + \frac{\kappa}{2}(\nabla\phi)^2 \tag{46}$$

where $\phi = \frac{\rho_A - \rho_B}{\rho_A + \rho_B}$ is the local order parameter (i.e. $\phi = +1$ for liquid A and $\phi = -1$ for liquid B) and a, b and κ are free parameters. Phase separation can occur when $a < 0$, and the surface tension is given by $\gamma = \sqrt{8\kappa a^3/9b^2}$. The chemical potential is the functional derivative of the free energy:

$$\mu = \frac{\delta \psi}{\delta \phi} = a\phi + b\phi^3 - \kappa \nabla^2 \phi. \tag{47}$$

More recently, hybrid approaches have emerged[114]. In order to understand their novelty, it is necessary to realise that any additional equation of motion beside the Navier-Stokes equation that is involved in the dynamics of the complex fluid can be solved as well with an LBM-style method by defining additional sets of either scalar, vectorial or tensorial distribution functions. The corresponding observables are then obtained by taking moments of these distribution function is just the same fashion. The equilibrium distributions are similar to Eq. 41, but have different coefficients owing to the different target equation. A good example of this full-LB approach has been given for the dynamics of liquid crystals[115], where a partial differential equation for the tensorial order parameter, often referred to as Q-tensor, needs to be solved on top of the Navier-Stokes equation. Full-LB approaches can feature increased stability compared to alternative methods. But on the down side they require more memory and floating point operations per timestep.

Hybrid schemes do not define additional sets of distribution functions and use LBM only for the Navier-Stokes part of the problem. Any partial differential equation is solved with finite-difference schemes. In the above example of liquid crystals and based on a D3Q19 LB model, this reduces the memory requirements by almost 79%: Instead of 19 distributions for the Navier-Stokes equation and 5×19 distributions for the 5 independent components of the Q-tensor in a full-LB approach, a hybrid model uses only 19 distributions for Navier-Stokes and holds the 5 independent components of the Q-tensor directly rather than reconstructing them from moments of the distribution. This, of course, is an extreme example due to the tensorial nature of the order parameter.

Like other models that involve phase boundaries or interfaces, both SC and FE models suffer from spurious currents in regions of large order parameter gradients. They can be reduced by higher-order isotropic schemes[116]. We refer to the relevant literature[85,86] for more details on multi-phase and multi-component mixtures.

4.4 Applications

Among the first soft matter applications of LBM were flows in porous media[82], particulate flows[117] and flows of polymer solvent systems[106]. These systems are notoriously difficult to model with standard CFD methods. On the contrary, LBM offers a comparably simple way of handling boundary conditions and fluid-solid interactions and seems almost ideally suited. For instance, the flow through a porous rock material, traditionally only describable through effective laws such as Darcy's law, can be simulated directly.

With regard to multi-component systems like binary[112] or amphiphilic mixtures[119] of immiscible fluids, LBM has as well a number of advantages. The modelling of droplet coalescence or necking phenomena usually requires expensive interface tracking algorithms that can trigger numerical instabilities. In LBM, these systems can be modelled with two sets of LB distribution functions, or with one set of distributions and a level set in hybrid methods. The position of the interface is then simply defined by a value of the level set or by the isosurface where the densities of both fluids have the same value. These models have also been successfully applied to nanoparticle-stabilised suspensions (so-called Pickering emulsions and bijels, Fig. 6)[118] and wetting phenomena[120].

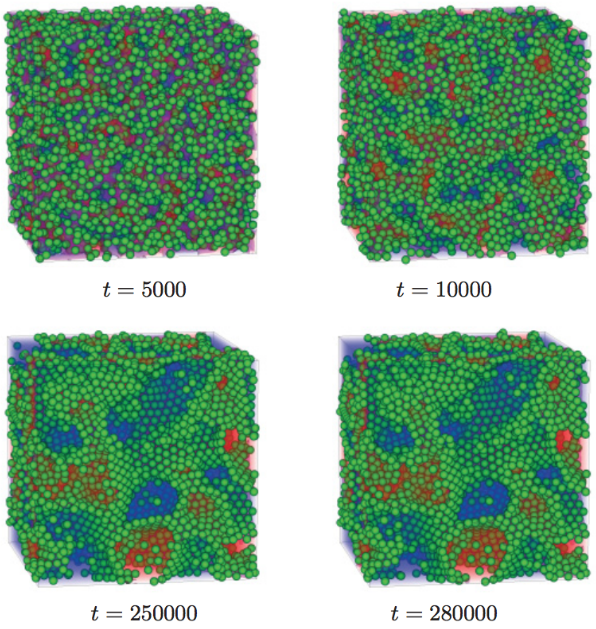

Fig. 6 3D visualisation of the a bicontinuous interfacially jammed emulsion gel (bijel). Shown are the particles and the two fluids, respectively. The two pictures at the bottom depict the bicontinuouity of the fluids and the attachment of the particles to the interface[118]. Reprinted figure with permission from Physical Review E. Copyright (2011) by the American Physical Society.

Free energy models and their ability to express thermodynamic forces on solute and solvent through gradients of chemical potentials come into their own when the composition and local structure of the complex fluid becomes increasingly intricate. This, for example, is the case in liquid crystals[115] where the anisotropic local order structure is described through a tensorial order parameter and gives rise to spatially varying rheological properties. Hybrid approaches in favour of full-LB schemes have been used for large-scale mesoscopic simulations of 3D liquid crystalline structures at very low shear rates[121]. Similar concepts have been used to study active liquid crystals and gels[114]. They can be described through the same methodology with modified free energy functionals that account for the active components and additional forces.

Other interesting applications comprise charged soft matter systems and electrokinetic phenomena. An important step forward was the development of the link-flux method[122] by which the leakage of charge between interfaces and surfaces could be prevented. The scheme has been used in the mesoscopic study of the electrophoresis of charged colloids[123] and polyelectrolytes[124]. Recently, the scheme has been improved to reduce spurious currents[125].

5 High-Performance Computing

The physics of many soft condensed matter systems can be investigated without ever considering large-scale simulations. However, high-performance computing (HPC) can be of particular importance, not only for simulation methodologies that can capture the long-range nature of hydrodynamic interactions — one focus of this tutorial review. Soft matter generally shows a tendency to self-organise into characteristic mesoscopic structures that in turn determine its macroscopic properties. Multiscale modelling techniques, accompanied by the appropriate computing power, are finding increasingly wide-spread use in this field of research.

The so-called rise of the machines has indeed transformed HPC from an exotic niche technology into an indispensable tool for science and engineering. In fact, since the beginning of this millennium, the computing power of the number one facility listed in the bi-annually published TOP500 has increased by more than a factor 40,000. With the advent of even more powerful graphics cards and many-core processors, an end to this astonishing development is not in sight.

At the same time, HPC architectures are rapidly diversifying and become more complex. This makes it even more difficult to program and harvest their computational power in the future. This development began about a decade ago and is directly linked to the breakdown of the so-called Dennard scaling. As the size and capacitance of integrated circuits decreased (and continues to decrease according to Moore's law), Dennard scaling allowed to operate them as well at lower voltages. Hence, the power gains in terms of lower capacitance and operating voltage could be "invested" in a higher clock frequency, making processors ever faster whilst keeping the power consumption more or less constant. This design principle became unviable as currents in the devices have weakened and the operating voltage in the devices cannot be further reduced. Therefore, the faster processors can only be created through more compute cores that run calculations in parallel.

In this paragraph, we will glance at the latest developments in the area of accelerators and coprocessors and describe the requirements that these increasingly heterogeneous computing architectures pose to the programmer. Finally, we will name a few examples of production codes used for soft matter research.

5.1 General Purpose Graphics Processing Units

General purpose graphics processing units (GPGPUs), also known as graphics cards or GPUs, were originally designed to handle the large amount of floating-point operations that occur in the graphics calculations of computer games. They made their first appearance in the world of scientific computing around 2007. Back then, Nvidia introduced its compute unified device architecture (CUDA), an application programming interface (API) that vastly simplified the programming of GPUs. This has so far led to a dominance of the Nvidia framework in the area of scientific computing with GPUs and prevented other vendor-independent standards like the open computing language (OpenCL) from attaining the same level of maturity. Alternative higher-level directive-based standards like OpenACC exist and permit automatic handling of data management and computational offloading at the expense of slightly reduced performance due to reduced control over data movements. This trade-off, however, makes more than up for the simplicity and portability as OpenACC-enabled code can run as well on many-core processors.

At the time of writing, the latest generation of Nvidia's Pascal architecture-based Tesla P100 GPUs delivers a peak performance of around 5 TFLOPs (5×10^{12} double-precision floating point operations per second) on 3584 CUDA cores at a power input of around 300 W, or 3–4 laptops. To put this in perspective, the fastest supercomputer in November 2000 was the ASCI White at Lawrence Livermore National Laboratory which achieved about the same performance with a power input of 3 MW (and another 3 MW for cooling, adding up to an electricity bill of roughly \$6 million p.a.). The Tesla P100 also features 500+ GB/s memory bandwidth and a unified memory space to which both GPU and CPU can point. This alleviates the burden of having to copy data back and forth between CPU (host) and GPU (device), a process that uses the traditionally slowest part in a computer and came always at significant costs.

However, in order to unleash the power of the GPU, algorithmic parallelism has to be fully exposed and mapped to the architectural parallelism of the hardware on which the code is deployed. The so-called task or thread-level parallelism (TLP) that modern accelerators offer through their hundreds or thousands of compute cores working in parallel is only one way to expose parallelism. Another concept, which dates back to the early days of vector processors, is known as data parallelism. The basic idea behind it is that one single instruction processes a chunk of similar data elements at the same time. Today, virtually all types of processors, also multicore CPUs, use advanced vector extensions (AVX) which are extensions to the x86 instruction set. These are also known as SIMD (Single Instruction Multiple Data) instructions. Obtaining a good performance on any kind of accelerator, co-processor or many-core chip depends increasingly on whether data parallelism can be exploited. Quite often this necessitates a complete redesign of the memory layout and memory access pattern, which can be prohibitive in the case of big legacy codes.

5.2 Intel Xeon Phi

While GPGPUs used to have a commanding lead over alternative concepts of co- and many-core processors, the latest edition of Intel's many integrated core (MIC) architecture, the Xeon Phi Knights Landing (KNL) has stirred up some competition. It consists of a smaller number of compute cores, typically 72, which can be run in a hyper-threading mode, giving in total up to 288 threads. The peak performance is around 3.5 TFLOPs at an energy input comparable to Nvidia's Tesla P100. The memory bandwidth of the KNL has been significantly increased compared to older generations; it is at about 400+ GB/s.

The KNL gets its performance from SIMD instructions on very large vector units. Hence, without exposing data parallelism, the performance will be noticeably degraded. In fact, data parallelism is probably even more crucial for MICs than for GPUs. That said, obtaining a consistently good performance with these devices, GPUs or MICs coprocessors alike, when real-world problems rather than simplistic benchmarks are considered, remains generally challenging. Performance measurements gained with one and the same code can vary substantially depending on the individual scientific problem.

Nevertheless, MICs offer an unparalleled advantage over GPUs in terms of portability. In order to port a code to the GPU, it is often necessary to rewrite large parts. This frequently leads to more complex functionality and much longer source code compared to its CPU counterparts. Once the investment has been made, the performance gain can be impressive. But the code can be only deployed on GPUs, or when CUDA has been used even only on Nvidia GPUs.

Code written in standard programming languages such as C, C++ and Fortran using the most common abstractions for parallel programming, i.e., message passing interface (MPI) for distributed memory or OpenMP and even OpenACC for shared memory architectures, can be compiled for the Xeon Phi coprocessor. Hence, the code can run on a variety of architectures, ranging from multi-core CPUs in laptops over workstations, many-core processors and computing clusters to supercomputers with on-node co-processors. Although to obtain a good performance the same fundamental principles apply as for the GPU (exposing sufficient data parallelism and providing a cache-coherent memory layout and access pattern), the entry barrier is much lower and the portability significantly higher. Moreover, performance improvements made for the Xeon Phi will equally benefit when the code is run on simple multicore CPUs and vice versa.

5.3 Community Codes

Developing efficient software for HPC applications has never been a trivial task as it normally exceeds the resources of a typical research group. Moreover, the pool of skilled programmers who have also sufficient scientific experience in their area of research is limited. This makes it difficult to conduct ambitious code projects, even when the funding is in place! Hence, it is not surprising that community codes have emerged which are sometimes used by thousands of researchers across the world. Here, we will only mention open source codes that are freely available for academic users. The list below can only be a cross section and does not claim to be comprehensive.

A prime example of community code is the large-scale atomic/molecular massively parallel simulator (LAMMPS)[126], developed by Sandia National Laboratories, USA. LAMMPS is modular and relatively easy to extend, which makes it an attractive code to base individual research projects on. Its latest version features a vast number of models, particle types, force fields, ensembles and integrators. In fact, to our knowledge LAMMPS is the only code that contains an implementation of virtually all methods mentioned in this review. That said, each method comes naturally in a multitude of different flavours and modifications. The LAMMPS implementation contains only some of these features. The MPC model, for instance, uses the stochastic rotation dynamics version, whereas the LB model is an implementation of Ollila's above mentioned algorithm. The existing implementations form an excellent starting point for further development of LAMMPS, which is one of the reasons why new features are constantly being added to every new release. LAMMPS has specific strengths for coarse-grained modelling, making it suitable for soft matter research in general. The code shows very good performance up to millions of simulated particles and thousands of cores, and the number of multi-GPU- and many-core-enabled force fields and features is continuously growing.

Another highly versatile software package for performing many-particle molecular dynamics simulations, with special emphasis on coarse-grained models as they are used in soft matter research is ESPResSo (Extensible Simulation Package for RESearch on SOft matter)[127,128]. It is commonly used to simulate systems such as polymers, colloids, ferro-fluids and biological systems, for example DNA or lipid membranes. ESPResSo also contains a unique selection of efficient algorithms for treating Coulomb interactions. Recently, several grid based algorithms such as lattice Boltzmann and an electrokinetics solver have been added.

Another interesting code is HOOMD-blue[129], a general purpose multi-GPU enabled molecular dynamics simulation toolkit from the University of Michigan, USA. Besides Brownian, Langevin dynamics, DPD and a number of ensembles, the hard particle capabilities and a variety of shape classes for Monte Carlo simulations are particularly worth mentioning. Other important community codes are GROMACS[130] and NAMD[131] which are popular in the biomolecular community. They are both highly optimised and perform exceptionally well in term of parallel efficiency, but lack the general extensibility that for instance LAMMPS offers.

Excellent parallel performance is perhaps not trivial, but much easier to achieve for lattice Boltzmann codes. The algorithm requires only communication between nearest neighbours on a regular grid and is intrinsically parallel — unlike MD-type algorithms, where only the introduction of cutoffs, neighbour lists and sophisticated communication patterns renders the problem parallelisable. Nevertheless, a true LBM community code has so far not emerged. However, both OpenLB[132], developed at the Karlsruhe Institute of Technology, Germany, and Palabos[133], the open source project developed in a collaboration between the University of Geneva and FlowKit Ltd. Lausanne, Switzerland, come probably closest to this endeavour. waLBerla[134], developed by the University of Erlangen-Nürnberg, is another LBM code with particular strengths in solving partial differential equations that are coupled to the fluid flow. Another open source LBM code specifically for soft matter is Ludwig[135]. It is being developed in a collaboration between different European project partners in the UK, France, Switzerland and Spain and features a wide range of mesoscopic models for complex fluids and active matter.

6 Conclusions and Outlook

This tutorial review gives an introduction to mesoscopic modelling and simulation of soft matter. We have focused on what we think are the three most popular methods and have reviewed the basic algorithm and features that are particularly relevant to mesoscale modelling. We have highlighted the specific strengths of each method with a few selected examples.

As noted in the introduction, the mesoscopic methods are based on conservation of mass, momentum and energy. They differ in the way the conservation laws are translated into a coarse-grained representation, resulting in different degrees of freedom and different equations of motion for each method. This also results in specific limitations and advantages in each case.

Dissipative Particle Dynamics (DPD), the method which is probably closest to traditional molecular dynamics, is an off-lattice method which has the capability to resolve hydrodynamic space and time scales significantly larger than those in traditional MD. The particles in DPD represent whole atoms or regions of solvent atoms and interact via effective forces which conserve momentum locally and deliver the correct hydrodynamic behaviour even for relatively low particle numbers. DPD integration algorithms slightly differ from conventional MD integrators and take into account the specific symmetry requirements in order to fulfil local momentum or global energy conservation. DPD is particularly well suited for specific physicochemical interactions on the particle-level, e.g., for systematic coarse-graining of solvation effects, heat conduction and convection in nano-materials, or the simulation of biological membranes, macromolecules or multiphase systems subject to different flow conditions and geometries.

Multi-particle collision dynamics (MPC) implements discrete streaming and collision steps to update the positions and velocities of an ensemble of fluid particles. The molecular interactions are coarse-grained by sorting the particles into collision cells and applying a simplified collision rule to exchange momentum. The collision cells also function as averaging volumes to calculate local hydrodynamic fields. This makes MPC amenable to a kinetic description from which the hydrodynamic Navier-Stokes equations emerge. In this sense, MPC bridges between the particle view and the kinetic view of the fluid. The possibility to tweak the collision rule such that different thermodynamic ensembles can be reproduced make MPC well suited to study transport phenomena under different ambient conditions. Heat transport due to temperature gradients and other non-equilibrium effects are important examples for the use of MPC.

The lattice Boltzmann method (LBM) is a lattice-based scheme that discretises time, space and velocity space. The commonly chosen discretisation leads to a simple and nearly local algorithm that naturally lends itself to parallel simulations. Due to its kinetic nature, complex and moving boundary conditions in the LBM are relatively straightforward to implement, when compared to conventional Navier-Stokes solvers. Since LBM operates with averaged particle distributions, it is especially suited for non-Brownian problems, e.g., non-colloidal particle suspensions, although fluctuations can be re-introduced. LBM is a popular method for multi-phase or multi-component problems, e.g., droplet dynamics or water/oil flow in porous media. Furthermore, it is straightforward to couple additional physics with the LBM algorithm, which enables applications such as liquid crystal dynamics or electrophoresis.

Interestingly, direct comparisons between the three above mentioned methods are scarce – at least we are not aware of any. This is perhaps a knowledge gap that the computational soft matter community may want to address in the future. Nevertheless, it is possible to draw a comparison between some more general aspects of each method.
A generic property of DPD is that both solute and solvent are modelled through coarse-grained particles that resemble each other closely. This characteristic feature makes the inclusion of specific solute-solvent interactions simple. These can also be modelled via effective coarse-grained force-fields that describe the averaged atomistic or molecular interactions on a larger length and time scale. Both DPD and MPC are fully thermalised methods and are natively connected to typical thermostatic or thermodynamic algorithms like Nosé-Hoover thermostats or Langevin dynamics. Whilst this is usually seen as an advantage, it allows only for fluctuating solutions of hydrodynamic problems. While fluctuations can be incorporated as well into LBM, arguably in a slightly more complicated manner, LBM permits primarily quasi noise-free, ballistic solutions. In form of the Chapman-Enskog expansion, LBM offers a tool which allows all specific modifications of the microscopic dynamics to be directly related to a macroscopic target equations, providing quasi- or near-analytical insight into physical problems.

Mesoscopic modelling continues to be an area of active research, and multiphase fluids and non-equilibrium soft matter offer a range of interesting research problems. Moreover, we anticipate that mesoscale methods will play a central role in enabling extreme-scale applications on future HPC systems. We hope that this tutorial review will help the reader to choose the appropriate methods to address the mesoscale physics that is relevant in their research problem.

Acknowledgments

The authors are indebted to Peter V. Coveney for his support and many valuable discussions. OH acknowledges support from the EPSRC Early Career Fellowship Scheme (EP/N019180/2). TK thanks the University of Edinburgh for the award of a Chancellor's Fellowship.

References

1 M. Doi, *Soft Matter Physics*, OUP, Oxford, 2013.
2 *The Oxford Handbook of Soft Condensed Matter*, ed. E. Terentjev and D. Weitz, OUP, Oxford, 2015.
3 J. Rote and S. Prager, *J. Chem. Phys.*, 1969, **50**, 4831–4837.
4 A. Ricci and G. Ciccotti, *Mol. Phys.*, 2003, **101**, 1927–1931.
5 D. L. Ermak and J. A. McCammon, *The Journal of Chemical Physics*, 1978, **69**, 1352–1360.
6 M. Fixman, *Macromolecules*, 1986, **19**, 1204–1207.
7 R. R. Schmidt, J. G. H. Cifre and J. G. de la Torre, *The Journal of Chemical Physics*, 2011, **135**, 084116.
8 T. T. Pham, U. D. Schiller, J. R. Prakash and B. Duenweg, *J. Chem. Phys.*, 2009, **131**, 164114.
9 A. J. Banchio and J. F. Brady, *J. Chem. Phys.*, 2003, **118**, 10323–10332.
10 G. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford University Press, Oxford, 1994.

11 A. Malevanets and R. Kapral, *J. Chem. Phys.*, 1999, **110**, 8605.

12 T. Ihle and D. M. Kroll, *Phys. Rev. E*, 2003, **67**, 066705.

13 C. A. Marsh and P. V. Coveney, *J. Phys. A: Math. Gen.*, 1998, **31**, 6561.

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

15 C.-C. Huang, A. Varghese, G. Gompper and R. G. Winkler, *Phys. Rev. E*, 2015, **91**, 013310.

16 J. M. Yeomans, *Physica A Stat. Mech. Appl.*, 2006, **369**, 159 – 184.

17 F. Toschi and S. Succi, *Europhys. Lett.*, 2005, **69**, 549.

18 J. T. Padding and A. A. Louis, *Phys. Rev. E*, 2006, **74**, 031402.

19 P. Hoogerbrugge and J. Koelman, *Europhys. Lett.*, 1992, **19**, 155.

20 J. Koelman and P. Hoogerbrugge, *Europhys. Lett.*, 1993, **21**, 363.

21 P. Español, *Phys. Rev. E*, 1995, **52**, 1734.

22 R. Groot and P. Warren, *J. Chem. Phys.*, 1997, **107**, 4423.

23 E. Moeendarbary, T. Y. Ng and M. Zangeneh, *Int. J. Appl. Mech.*, 2009, **1**, 737.

24 E. Moeendarbary, T. Y. Ng and M. Zangeneh, *Int. J. Appl. Mech.*, 2010, **2**, 161.

25 P. Español and P. Warren, *Europhys. Lett.*, 1995, **30**, 191.

26 P. Nikunen, M. Karttunen and I. Vattulainen, *Comput. Phys. Commun.*, 2003, 407–423.

27 P. Español, *Europhys. Lett.*, 1997, **40**, 631.

28 I. Pagonabarraga and D. Frenkel, *J. Chem. Phys.*, 2001, **115**, 5015.

29 E. Flekkoy and P. Coveney, *Phys. Rev. Lett.*, 1999, **83**, 1775.

30 E. Flekkoy, P. Coveney and G. De Fabritiis, *Phys. Rev. E*, 2000, **62**, 2140.

31 P. Español and M. Revenga, *Phys. Rev. E*, 2003, **67**, 026705.

32 C. Lowe and M. Dreischor, *Lect. Notes Phys.*, 2004, **640**, 35.

33 S. Nielsen, C. Lopez, G. Srinivas and M. Klein, *J. Phys. Condens. Matter*, 2004, **16**, R481.

34 M. B. Liu, G. R. Liu, L. W. Zhou and J. Z. Chang, *Arch. Computat. Methods Eng.*, 2015, **22**, 529–556.

35 P. Nikunen, I. Vattulainen and M. Karttunen, *Phys. Rev. E*, 2007, **75**, 036713.

36 J. Smiatek and F. Schmid, *Comput. Phys. Commun.*, 2011, **182**, 1941–1944.

37 S. R. Maduar, A. V. Belyaev, V. Lobaskin and O. I. Vinogradova, *Phys. Rev. Lett.*, 2015, **114**, 118301.

38 D. A. Fedosov, W. Pan, B. Caswell, G. Gompper and G. E. Karniadakis, *Proc. Natl. Acad. Sci. USA*, 2011, **108**, 11772–11777.

39 D. A. Fedosov, B. Caswell and G. E. Karniadakis, *Biophys. J.*, 2010, **98**, 2215–2225.

40 G. Gompper, T. Ihle, D. M. Kroll and R. G. Winkler, in *Advanced Computer Simulation Approaches For Soft Matter Sciences III*, ed. Holm, C and Kremer, K, Springer-Verlag Berlin, 2009, vol. 221 of Advances in Polymer Science, pp. 1–87.

41 T. Ihle and D. Kroll, *Phys. Rev. E*, 2001, **63**, 020201.

42 C.-C. Huang, G. Gompper and R. G. Winkler, *Phys. Rev. E*, 2012, **86**, 056711.

43 H. Noguchi and G. Gompper, *Phys. Rev. E*, 2008, **78**, 016706.

44 J. F. Ryder, *Ph.D. thesis*, University of Oxford, 2005.

45 E. Tüzel, M. Strauss, T. Ihle and D. M. Kroll, *Phys. Rev. E*, 2003, **68**, 036701.

46 T. Ihle and D. M. Kroll, *Phys. Rev. E*, 2003, **67**, 066706.

47 T. Ihle, E. Tüzel and D. M. Kroll, *Phys. Rev. E*, 2005, **72**, 046707.

48 N. Kikuchi, C. M. Pooley, J. F. Ryder and J. M. Yeomans, *J. Chem. Phys.*, 2003, **119**, 6388–6395.

49 C. M. Pooley and J. M. Yeomans, *J. Phys. Chem. B*, 2005, **109**, 6505–6513.

50 A. Lamura, G. Gompper, T. Ihle and D. M. Kroll, *Europhys. Lett.*, 2001, **56**, 319.

51 R. G. Winkler and C.-C. Huang, *The Journal of Chemical Physics*, 2009, **130**, 074907.

52 M. Hecht, J. Harting, T. Ihle and H. J. Herrmann, *Phys. Rev. E*, 2005, **72**, 011408.

53 T. Ihle, E. Tüzel and D. M. Kroll, *Europhys. Lett.*, 2006, **73**, 664.

54 T. Ihle and E. Tüzel, *Progr. Comput. Fluid Dyn.*, 2008, **8**, 138–155.

55 E. Tüzel, T. Ihle and D. M. Kroll, *Math. Comput. Simul.*, 2006, **72**, 232–236.

56 Y. Hashimoto, Y. Chen and H. Ohashi, *Comput. Phys. Commmun.*, 2000, **129**, 56–62.

57 T. Sakai, Y. Chen and H. Ohashi, *Comput. Phys. Commun.*, 2000, **129**, 75–81.

58 Y. Inoue, Y. Chen and H. Ohashi, *J. Chem. Phys.*, 2004, **201**, 191–203.

59 Y. Inoue, S. Takagi and Y. Matsumoto, *Comput. Fluids*, 2006, **35**, 971–977.

60 Y. G. Tao, I. O. Goetze and G. Gompper, *J. Chem. Phys.*, 2008, **128**, 144902.

61 S. Ji, R. Jiang, R. G. Winkler and G. Gompper, *J. Chem. Phys.*, 2011, **135**, 134116.

62 B. Kowalik and R. G. Winkler, *J. Chem. Phys.*, 2013, **138**, 104903.

63 R. G. Winkler, K. Mussawisade, M. Ripoll and G. Gompper, *J. Phys.: Condens. Matter*, 2004, **16**, S3941.

64 R. G. Winkler, A. Ripoll, K. Mussawisade and G. Gompper, *Comput. Phys. Commun.*, 2005, **169**, 326–330.

65 M. Ripoll, R. G. Winkler and G. Gompper, *Phys. Rev. Lett.*, 2006, **96**, 188302.

66 M. Ripoll, R. G. Winkler and G. Gompper, *Eur. Phys. J. E*, 2007, **23**, 349–354.

67 S. P. Singh, C.-C. Huang, E. Westphal, G. Gompper and R. G. Winkler, *J. Chem. Phys.*, 2014, **141**, 084901.

68 S. Y. Reigh, R. G. Winkler and G. Gompper, *Soft Matter*, 2012, **8**, 4363–4372.

69 J.-B. Fleury, U. D. Schiller, S. Thutupalli, G. Gompper and R. Seemann, *New J. Phys.*, 2014, **16**, 063029.

70 U. D. Schiller, J.-B. Fleury, R. Seemann and G. Gompper, *Soft Matter*, 2015, **11**, 5850–5861.

71 E. Lauga and T. R. Powers, *Rep. Prog. Phys.*, 2009, **72**, 096601.

72 J. Elgeti, R. G. Winkler and G. Gompper, *Rep. Prog. Phys.*, 2015, **78**, 056601.

73 S. Y. Reigh, R. G. Winkler and G. Gompper, *PLOS ONE*, 2013, **8**, e70868.

74 J. Hu, M. Yang, G. Gompper and R. G. Winkler, *Soft Matter*, 2015, **11**, 7867–7876.

75 M. Yang and M. Ripoll, *Soft Matter*, 2013, **9**, 4661–4671.

76 M. Yang and M. Ripoll, *Phys. Rev. E*, 2011, **84**, 061401.

77 D. Lüsebrink and M. Ripoll, *J. Chem. Phys.*, 2012, **136**, 084106.

78 M. Yang, A. Wysocki and M. Ripoll, *Soft Matter*, 2014, **10**, 6208–6218.

79 M. Yang and M. Ripoll, *Soft Matter*, 2014, **10**, 1006–1011.

80 M. Yang, R. Liu, M. Ripoll and K. Chen, *Nanoscale*, 2014, **6**, 13550–13554.

81 M. Yang, M. Ripoll and K. Chen, *J. Chem. Phys.*, 2015, **142**, 054902.

82 S. Chen and G. Doolen, *Annu. Rev. Fluid Mech.*, 1998, **30**, 329.

83 S. Succi, *Lattice Boltzmann Equation: For Fluid Dynamics and Beyond*, Oxford University Press, New York, 2001.

84 C. Aidun and J. Clausen, *Annu. Rev. Fluid Mech.*, 2010, **42**, 439.

85 Z. Guo and C. Shu, *Lattice Boltzmann Method and its Application in Engineering*, World Scientific, New York, 2013.

86 T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva and E. M. Viggen, *The Lattice Boltzmann Method: Principles and Practice*, Springer, 2016.

87 U. Frisch, B. Hasslacher and Y. Pomeau, *Phys. Rev. Lett.*, 1986, **56**, 1505–1508.

88 G. McNamara and G. Zanetti, *Phys. Rev. Lett.*, 1988, **61**, 2332.

89 X. Shan, X.-F. Yuan and H. Chen, *J. Fluid Mech.*, 2006, **550**, 413–441.

90 Y. Qian, D. d'Humières and P. Lallemand, *Europhys. Lett.*, 1992, **17**, 479.

91 D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand and L. Luo, *Phil. Trans. Roy. Soc. London A*, 2002, **360**, 437.

92 B. Dünweg and A. J. C. Ladd, in *Advanced Computer Simulation Approaches For Soft Matter Sciences III*, ed. Holm, C and Kremer, K, Springer-Verlag Berlin, 2009, vol. 221 of Advances in Polymer Science, p. 89.

93 I. Ginzburg, F. Verhaeghe and D. d'Humières, *Commun. Comput. Phys.*, 2008, **3**, 427–478.

94 S. Ansumali, I. Karlin and H. C. Ottinger, *Europhys. Lett.*, 2003, **63**, 798.

95 Z. Guo, C. Zhen and B. Shi, *Phys. Rev. E*, 2002, **65**, 046308.

96 S. S. Chikatamarla and I. V. Karlin, *Phys. Rev. Lett.*, 2006, **97**, 190601.

97 P. Dellar, *J. Comput. Phys.*, 2014, **259**, 270.

98 R. Adhikari, K. Stratford, M. Cates and A. Wagner, *Europhys. Lett.*, 2005, **71**, 473.

99 M. Gross, R. Adhikari, M. E. Cates and F. Varnik, *Phys. Rev. E*, 2010, **82**, 056714.

100 R. Cornubert, D. d'Humières and D. Levermore, *Physica D*, 1991, **47**, 241–259.

101 I. Ginzbourg and P. M. Adler, *J. Phys. II France*, 1994, **4**, 191–214.

102 A. J. C. Ladd, *J. Fluid Mech.*, 1994, **271**, 285–309.

103 C. K. Aidun and Y. Lu, *J. Stat. Phys.*, 1995, **81**, 49–61.

104 M. Bouzidi, M. Firdaouss and P. Lallemand, *Phys. Fluids*, 2001, **13**, 3452–3459.

105 I. Ginzburg and D. d'Humières, *Phys. Rev. E*, 2003, **68**, 066614.

106 P. Ahlrichs and B. Dünweg, *Int. J. Mod. Phys. C*, 1998, **09**, 1429–1438.

107 P. Bagchi, *Biophys. J.*, 2007, **92**, 1858–1877.

108 C. S. Peskin, *Acta Numerica*, 2002, **11**, 479–517.

109 S. T. Ollila, C. J. Smith, T. Ala-Nissila and C. Denniston, *Multiscale Modeling & Simulation*, 2013, **11**, 213–243.

110 X. Shan and H. Chen, *Phys. Rev. E*, 1993, **47**, 1815.

111 M. Swift, W. Osborn and J. Yeomans, *Phys. Rev. Lett.*, 1995, **75**, 830.

112 M. Swift, E. Orlandini, W. Osborn and J. Yeomans, *Phys. Rev. E*, 1996, **54**, 5041.

113 A. J. Briant and J. M. Yeomans, *Phys. Rev. E*, 2004, **69**, 031603.

114 D. Marenduzzo, E. Orlandini, M. Cates and J. Yeomans, *Phys. Rev. E*, 2007, **76**, 031921.

115 C. Denniston, D. Marenduzzo, E. Orlandini and J. Yeomans, *Phil. Trans. R. Soc. Lond. A*, 2004, 1745–1754.

116 K. Connington and T. Lee, *J. Mech. Sci. Technol.*, 2012, **26**, 3857.

117 A. Ladd and R. Verberg, *J. Stat. Phys.*, 2001, **104**, 1191.

118 F. Jansen and J. Harting, *Phys. Rev. E*, 2011, **83**, 046707.

119 H. Chen, B. Boghosian, P. Coveney and M. Nekovee, *Proc. R. Soc. A*, 2000, **456**, 2043–2057.

120 M. Blow and J. Yeomans, *Langmuir*, 2010, **26**, 16071–16083.

121 O. Henrich, K. Stratford, P. Coveney, M. Cates and D. Marenduzzo, *Soft Matter*, 2013, 10243–10256.

122 F. Capuani, I. Pagonabarraga and D. Frenkel, *J. Chem. Phys.*, 2004, 973–986.

123 G. Giupponi and I. Pagonabarraga, *Phys. Rev. Lett.*, 2011, 248304.

124 K. Grass and C. Holm, *Farad. Discuss*, 2010, 57–70.

125 G. Rempfer, G. Davies, C. Holm and J. de Graaf, *J. Chem. Phys.*, 2016, 044901.

126 S. Plimpton, *J. Comput. Phys.*, 1995, 1–19.

127 H. J. Limbach, A. Arnold, B. A. Mann and C. Holm, *Comput. Phys. Commun.*, 2006, **174**, 704–727.

128 A. Arnold, O. Lenz, S. Kesselheim, R. Weeber, F. Fahrenberger, D. Roehm, P. Košovan and C. Holm, Meshfree Methods for Partial Differential Equations VI, 2013, pp. 1–23.

129 J. Anderson, C. Lorenz and A. Travesset, *J. Comput. Phys.*, 2008, 5342–5359.

130 M.J. Abraham, D. van der Spoel, E. Lindahl, et al., GROMACS User Manual version 2016.3, 2017, url: http://www.gromacs.org.

131 R. Bernardi, M. Bhandarkar, A. Bhatele, et al., NAMD User's Guide 2.12, 2016, url: http://http://www.ks.uiuc.edu.

132 M. J. Krause, T. Henn, A. Mink, et al., OpenLB Release 1.0: Open Source Lattice Boltzmann Code, online, 2016, url: http://www.openlb.net.

133 B. Chopard, D. Kontaxakis, D. Lagrava, et al., Palabos Version 1.5 Release 1, online, 2015, url: http://www.palabos.org.

134 C. Godenschwager, F. Schornbaum, M. Bauer, et al., Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis 2013 (SC13), url: http://www.walberla.net.

135 K. Stratford, et al., Ludwig, Soft Matter Simulation Software, online, 2017, url: http://ludwig.epcc.ed.ac.uk.