Mesoscopic Simulations for Problems with Hydrodynamics, with Emphasis on Polymer Dynamics

Burkhard Dünweg

Max Planck Institute for Polymer Research
Ackermannweg 10, D–55128 Mainz, Germany
duenweg@mpip-mainz.mpg.de

Summary. The contribution discusses two central models for polymer dynamics, the Rouse model and the Zimm model. The latter takes into account hydrodynamic interactions, and is hence appropriate for dilute solutions. In dense melts, the hydrodynamic interactions are screened, and the Rouse model is applicable (within limitations) as long as the chains are short enough to preclude reptation. The physics of this screening is discussed. The lecture then focuses on the methodological issue how to take hydrodynamic interactions into account in computer simulations. So far, the most successful methods are hybrid approaches where standard Molecular Dynamics for the polymer system is coupled to a mesoscopic model for momentum transport in the solvent. The most popular mesoscopic models are lattice Boltzmann and Dissipative Particle Dynamics. These methods are briefly discussed and contrasted. We describe a recent application of such an approach to the problem of hydrodynamic screening.

1 Polymer Dynamics

1.1 Overview

In this lecture, we will deal with the dynamic behavior of polymers [1] and discuss the simplest systems only: We will focus on *linear*, *flexible*, and *uncharged* macromolecules, disregading polydispersity (i. e. the broad distribution of chain lengths which usually occur in real systems), and study them in the *bulk*, in (or near) *thermal equilibrium*. We look at the systems from the point of view of coarse–grained models: We are not interested in the dependence of the properties on the details of the local chemistry, but rather ask for universal scaling laws. The aim of computer simulations within this sub–field of polymer physics is to carefully check the pertinent predictions, and to try to elucidate the underlying physics. For certain properties, simulations can be numerically much more accurate than experiments, and they therefore complement the latter.

Although our point of view and our restriction on the physical conditions may look quite narrow, there is nevertheless a rich host of phenomena which need explanation. Since we look at *solutions*, we can ask for the dependence of static and dynamic properties on *monomer concentration* c, *chain length* N (i. e. the number of monomers per chain), and the solvent quality. Usually polymer and solvent are the more miscible the higher the temperature is, i. e. the solvent quality can be parameterized in terms of temperature T — although there exist some cases where miscibility occurs at low temperatures, such that the system unmixes upon heating. Disregarding such "pathological", entropy–driven situations, Fig. 1.1 shows the generic phase diagram of such a solution.

One sees that even the statics is quite non–trivial, giving rise to a host of scaling laws and various crossovers, and that the dynamics is even more complex. Our present state of understanding is based on three fundamental models: The Rouse model [2], the Zimm model [3], and the reptation model [1]. In the present lecture we will discuss the Rouse and the Zimm model (see below). Reptation, which is characterized by curvilinear motion, and occurs in dense long–chain systems, does not play a role for the simulations to be described later, and shall hence not be discussed here.

So far, simulations have successfully treated the following cases:

- A single chain in good solvent with Zimm dynamics [4–10].
- A dense melt and its crossover from Rouse to reptation dynamics [11–14].
- Semidilute solutions characterized by a crossover from Zimm to Rouse dynamics as the concentration is increased [15].

Quite promising attempts have also been made to study the dynamics of the theta transition (i. e. a single chain collapses upon decrease of the solvent quality) [16–18]. Nevertheless, a systematic exploration of the plane concentration vs. solvent quality has not yet been done. After reading this article, the reader may perhaps understand why. For this reason, the present lecture will also disregard solvent quality effects and focus on good solvents (high temperatures in Fig. 1.1) only. The methods to be discussed in this lecture aim at an optimal exploitation of the different physical nature of the solvent and the solute, and this holds for any solvent quality.

To summarize: Our concern is to use computer simulations to put dynamic scaling laws in dilute, semidilute, and concentrated systems under scrutiny. We will take the underlying static scaling laws for granted (i. e. checked previously by other simulations and / or experiments). Nevertheless, in order to understand the dynamic models, it is necessary to first briefly review the statics.

1.2 Static Scaling in Polymer Solutions

The static conformations of flexible polymer chains are described via the statistics of a random coil [19]. We model the chain as a sequence of N

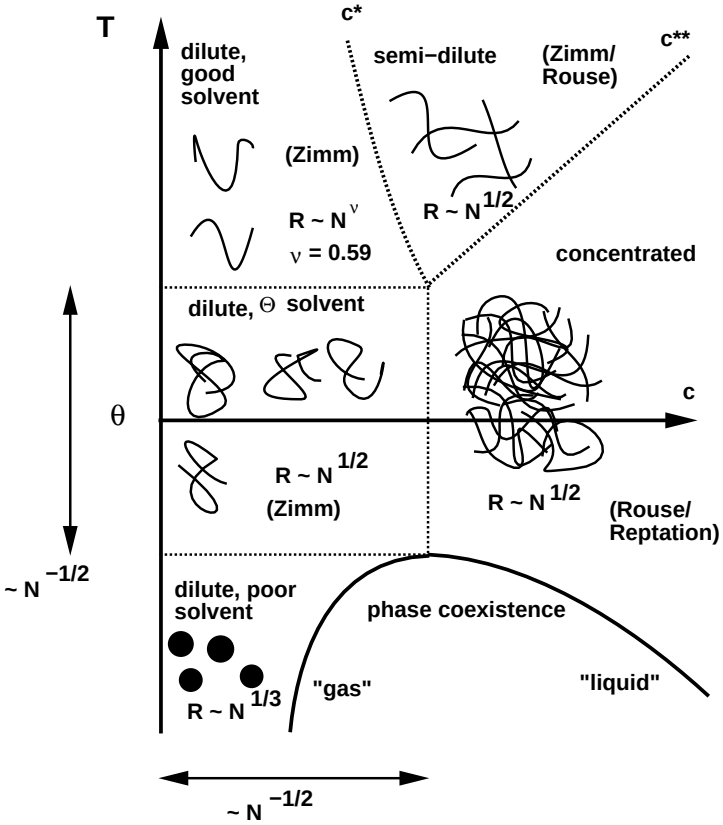

Fig. 1. Phase diagram of a polymer solution, in the plane monomer concentration c vs. temperature T (parameterizing solvent quality). The static properties are characterized by the scaling laws which describe the dependence of the chain size R (like the gyration radius or the end–to–end–distance) on the degree of polymerization N. In the dilute limit $c \to 0$, the so–called theta transition occurs, where at $T = \Theta$ single isolated chains collapse from a swollen random coil to a compact globule. For finite chain length N, this transition is "smeared out" over a temperature region $\Delta T \propto N^{-1/2}$, in which the chain conformations are Gaussian. Below Θ, phase coexistence between a "gas" of globules and a "liquid" of strongly interpenetrating Gaussian chains occurs. The corresponding critical point occurs at a very low concentration, $c_c \propto N^{-1/2}$, and in the vicinity of Θ, $\Theta - T_c \propto N^{-1/2}$. The crossover region which connects the regime of swollen isolated coils with that of the concentrated (Gaussian) solution at high temperatures is called the semidilute regime. The dynamics is characterized by the Zimm model in the dilute limit where hydrodynamic interactions are important, and by the Rouse model for dense systems where they are screened. For very dense systems and / or sufficiently long chains, where curvilinear motion dominates, the Rouse model must be replaced by the reptation model (or the crossover behavior between these two cases). Rouse and Zimm model are dexcribed in the text in detail. The Zimm–Rouse crossover which occurs in the semidilute regime is a central topic of the present lecture.

"monomers" with positions $\boldsymbol{r}_i$, $i = 1, \ldots, N$. From the chemist's point of view, a monomer is one chemical repeat unit, usually comprising several atoms (for instance, the repeat unit of polyethylene consists of one carbon and two hydrogen atoms). The position $\boldsymbol{r}_i$ can thus be viewed as the center of mass of the repeat unit i (or some similar quantity characterizing where the unit is). We denote the typical bond length with b,

$$\left\langle b^2 \right\rangle = \left\langle (\boldsymbol{r}_{i+1} - \boldsymbol{r}_i)^2 \right\rangle . \tag{1}$$

It is however also possible to combine several repeat units into a "super–monomer". In such a case, the same chain would be described by a larger value of b, and, correspondingly, by a smaller value of N. The notion of a monomer is therefore somewhat arbitrary. This leads us to the important principle of *scale invariance*: The measurable large–scale properties of the chain may not depend explicitly on the way in which the chain has been decomposed into monomers. By iterating this coarse–graining procedure ("renormalization group"), and applying dimension arguments, it is then easy to show that scale invariance implies power–law behavior. For a single chain which has no other important length scale than R (size of the chain as a whole), and b_{min} (shortest atomistic length scale below which a yet finer decomposition is impossible), the power law reads

$$R \sim b N^{\nu}. \tag{2}$$

Here we have assumed that one specific definition of a monomer (implying one specific value of N) has been chosen. The exponent ν depends on the physical conditions: For isolated chains in good solvent, the chains are swollen as a result of the excluded–volume interaction, with $\nu \approx 0.588$ in three dimensions. The statistics is that of a self–avoiding walk (SAW). In dense systems, the excluded volume interaction is screened [19, 20], hence ν has the Gaussian or random walk (RW) value $\nu = 1/2$. In what follows, the letter ν will either denote both values (in cases where the distinction between RW and SAW does not matter), or the SAW value (in cases where it does).

We have not specified precisely how to measure R; the scaling law applies to *all* ways of defining it. Convenient measures are the end–to–end–distance,

$$\left\langle R_E^2 \right\rangle = \left\langle (\boldsymbol{r}_N - \boldsymbol{r}_1)^2 \right\rangle , \tag{3}$$

the gyration radius

$$\left\langle R_G^2 \right\rangle = \frac{1}{N} \sum_i \left\langle (\boldsymbol{r}_i - \boldsymbol{R}_{CM})^2 \right\rangle \tag{4}$$

($\boldsymbol{R}_{CM}$ denoting the chain's center of mass, $\boldsymbol{R}_{CM} = N^{-1} \sum_i \boldsymbol{r}_i$), and the hydrodynamic radius

$$\left\langle \frac{1}{R_H} \right\rangle = \frac{1}{N^2} \sum_{i \neq j} \left\langle \frac{1}{r_{ij}} \right\rangle , \tag{5}$$

where $r_{ij} = |\boldsymbol{r}_i - \boldsymbol{r}_j|$.

Another important way to characterize the conformations is via the single–chain static structure factor $S(k)$, defined as

$$S(k) = \frac{1}{N} \left\langle \left| \sum_i \exp\left(i\boldsymbol{k} \cdot \boldsymbol{r}_i\right) \right|^2 \right\rangle = \frac{1}{N} \left\langle \sum_{ij} \exp\left(i\boldsymbol{k} \cdot \left(\boldsymbol{r}_i - \boldsymbol{r}_j\right)\right) \right\rangle . \quad (6)$$

On length scales $b \ll k^{-1} \ll R$, $S(k)$ does not depend on N (note that the addition of exponentials is just a RW in the complex plane). On the other hand, $S(k)$ must have the scaling form $S(k) = Nf(kR)$. This implies a power law decay $S(k) \propto k^{-1/\nu}$.

The conformations can always be described in terms of an effective potential V (potential of mean force) such that the equilibrium probability density is

$$P\left(\{\boldsymbol{r}_i\}\right) \propto \exp\left(-\frac{V\left(\{\boldsymbol{r}_i\}\right)}{k_B T}\right), \quad (7)$$

k_B and T denoting the Boltzmann constant and the absolute temperature, respectively.

For a Gaussian chain, V is just a harmonic potential:

$$V = \frac{3k_B T}{2b^2} \sum_{i=1}^{N-1} \left(\boldsymbol{r}_{i+1} - \boldsymbol{r}_i\right)^2 . \quad (8)$$

For a SAW chain, additional repulsive potentials between the monomers must be added in order to model the excluded–volume interaction. Models of this type are called "bead–spring" models.

Let us now discuss the crossover from SAW to RW behavior when the concentration is increased. We will always assume good solvent conditions. The coil size remains independent of concentration as long as the chains do not overlap. The concentration $c^\star$ where overlap starts to happen [19] is estimated via the requirement that an arrangement of unperturbed SAWs is just space–filling:

$$c^\star \sim \frac{N}{R^3} \sim \frac{N}{b^3 N^{3\nu}} \sim b^{-3} N^{-(3\nu-1)}. \quad (9)$$

Solutions with concentration $c \gg c^\star$ (well above overlap) but $c \ll b^{-3}$ (i. e. monomer concentration still very small) are called *semi–dilute*. Such solutions have another important length scale ξ, intermediate between b_{min} and R. ξ is called the "blob size" and can be defined as follows: If the chains were cut into sub–chains, each with size ξ, then the solution would be just at the overlap concentration corresponding to this lower molecular weight. On length scales below ξ, the statistics corresponds to SAW behavior, while for length scales beyond ξ RW behavior applies. Denoting the number of monomers within the blob with n, we have $\xi \sim bn^\nu$, and $c \sim n/\xi^3$, hence

$$\xi \sim b\left(cb^3\right)^{-\frac{\nu}{3\nu-1}} \propto c^{-0.77}. \quad (10)$$

The chain is then viewed as a RW sequence of blobs, with $R \sim \xi(N/n)^{1/2}$. The single–chain structure factor decays as $S(k) \propto k^{-1/\nu}$ for $b \ll k^{-1} \ll \xi$, and as k^{-2} for $\xi \ll k^{-1} \ll R$.

1.3 Rouse Model

The Rouse model [1, 2] is the simplest model of polymer dynamics, while the Zimm model and the reptation model are slightly more complicated modifications. Essentially, the Rouse model considers a single chain described by effective interactions as outlined in the previous paragraph, and its overdamped Brownian motion resulting from its coupling to a simple viscous background and to thermal noise. The Zimm model [3] replaces the viscous background by a hydrodynamic continuum which can transport momentum, and thus takes hydrodynamic interactions into account. Conversely, the reptation model [1] is a generalization of the Rouse model where topological constraints are taken into account in terms of an effective tube to which the chain is confined. In what follows, we will first outline the mathematical description of the Rouse model, and then attempt to critically assess its assumptions. In particular, we will try to briefly discuss the neglect of hydrodynamics (Zimm model physics), and the neglect of entanglements (reptation model physics). This discussion will be followed by an outline of the consequences of the Rouse equation of motion.

The dynamic variables of the Rouse model are just the positions of the monomers of a single test chain, $\boldsymbol{r}_i$, $i = 1, \ldots, N$. Taking into account that the model is supposed to describe its motion in an environment of other chains (a melt), we see that there is a colossal reduction in the number of degrees of freedom — solving the full many–body problem would require the positions and momenta of all monomers. Now, the Rouse model assumes the following overdamped Langevin equation of motion:

$$\frac{d}{dt}\boldsymbol{r}_i = \frac{1}{\zeta}\boldsymbol{F}_i + \boldsymbol{\rho}_i, \tag{11}$$

where ζ is the monomer friction coefficient, $\boldsymbol{\rho}_i$ the random displacement (per unit time) acting on monomer i, and $\boldsymbol{F}_i$ the force acting on monomer i, derived from the effective potential V (see previous subsection):

$$\boldsymbol{F}_i = -\frac{\partial V}{\partial \boldsymbol{r}_i}. \tag{12}$$

The random displacements are Gaussian white noise satisfying the standard fluctuation–dissipation theorem,

$$\langle \rho_i^\alpha \rangle = 0 \qquad \left\langle \rho_i^\alpha(t)\rho_j^\beta(t') \right\rangle = 2\frac{k_B T}{\zeta}\delta_{ij}\delta_{\alpha\beta}\delta(t-t'), \tag{13}$$

such that the equilibrium distribution produced by the Langevin process is the correct one. (Greek letters denote Cartesian indices.) The stochastic displacements in different directions, and of different monomers, are assumed as statistically independent.

At this point, it is clear that many important aspects have been disregarded. Firstly, the neglect of the momenta (i. e. that fact that only positions occur as dynamical variables) is safe only at the first, but not at second glance. The motivation of the neglect is the idea that in a dense simple fluid the motion of particles is essentially an oscillation in a local cage, and escape occurs only after many collisions, such that the memory of the initial momentum is completely lost on the time scale on which a monomer moves its own size. However, since the discovery of the long–time tails [21] we know that this is not quite true: Since momentum is a conserved quantity, it can only be transported away, but not simply destroyed. Let us therefore discuss the physics of momentum transport in some more detail. On long time and length scales, it can be described by the Stokes equation (hydrodynamic equation of motion for an incompressible fluid, where the nonlinear term is neglected):

$$\rho \frac{\partial}{\partial t} \boldsymbol{u} = \eta \nabla^2 \boldsymbol{u}, \qquad \nabla \cdot \boldsymbol{u} = 0, \tag{14}$$

where ρ is the fluid density, η its viscosity, and $\boldsymbol{u}$ the velocity flow field. The momentum transport hence takes place in a *diffusive* fashion, where the so–called "kinematic viscosity" $\eta_{kin} = \eta/\rho$ plays the role of a diffusion constant. Within the time t, an initial momentum therefore spreads into a sphere whose radius is of order $(\eta_{kin} t)^{1/2}$, or whose volume is of order $(\eta_{kin} t)^{3/2}$. This is the reason for the $t^{-3/2}$ decay of the velocity autocorrelation function. It should also be noted that a hydrodynamic description of a fluid is *always* valid on sufficiently large length and time scales for *any* fluid (including polymer melts).

These considerations, however, generate a puzzle for the validity of the Rouse model: Shouldn't one expect, from the physics of the long–time tails, that the monomer will need time "forever" until it really forgets its original velocity? Doesn't that imply that the description in terms of a position–only Langevin equation is expected to fail, and shouldn't one rather introduce an appropriate memory function to describe such a lack of forgetfulness?

However, let us look at the systems for which the Rouse model works (at least to a good approximation). These are short–chain melts. In such a system, the collisions between monomers do not occur in a nice and orderly fashion as in a simple fluid. Rather, the typical collision process is a chain–chain collision, such that an incoming kick will mainly result in a chain elongation against the connectivity forces, rather than being transported straight along. These processes destroy the memory to a large extent, and they are also the reason for hydrodynamic screening (see below). The random arrangement of chains results in a randomization of the scattering, and correlations are removed. Therefore, the monomer *does* ("at third glance") forget its initial

velocity rather quickly — except for (i) an extremely small contribution to the velocity autocorrelation function which comes from the long–time, large–length hydrodynamics of the melt (on scales beyond the gyration radius and the chain relaxation time), and can safely be neglected (it has so far never been resolved in simulational or experimental studies of melts), and (ii) a long–time negative contribution which describes the slowing down due to the spreading of correlations along the chain backbone [22]. However, phenomenon (ii) is actually faithfully described *within* the Rouse model. What this means is that the Rouse model monomer diffusion coefficient k_BT/ζ must be viewed as a *short–time* diffusion coefficient — not to be mistaken for the long–time diffusion coefficient, which is identical to that of the chain as a whole (i. e. much smaller) and which can be obtained as the time integral of the velocity autocorrelation function. Altogether, this argument restores the validity of the simple friction coefficient ansatz. Of course, the value of ζ depends on the definition of what one calls a monomer, just as the value of b does.

As a side remark, let us note that the typical "simple fluid" collision processes *will* play a role in a dilute solution, and the Rouse model is not expected to work. However, the hydrodynamic effects will not only induce correlations of the monomer with itself at later times, but also with *other* monomers on the chain. This phenomenon is called "hydrodynamic interaction". Taking these correlations into account, one obtains the Zimm model, which is discussed below in more detail. Considering again the single–monomer velocity autocorrelation function, one will have a short–time decay (giving rise to a short–time diffusion coefficient, or a monomer friction coefficient), followed by a negative contribution resulting from both the Rouse–like slowing down *and* long–time–tail memory.

As a second caveat of the Rouse equation of motion, it must be stressed that the model is a *single–chain* theory, i. e. all correlation effects with other chains are ignored. This latter neglect may be fine in dilute solution, but it is far from obvious for a dense melt where the chains are very close to each other. Indeed, the dramatic failure of the Rouse model for melts of sufficiently long chains, where rather the reptation model applies, is an obvious hint of this fact. The reptation model, where the entanglements with the other chains are replaced by a tube of a certain diameter [1], is of course yet another single–chain theory. From this point of view, it is not too surprising that the Rouse model does not work *precisely* for polymer melts. Rather, computer simulations and experiments have revealed a number of deviations [23, 24], the most interesting of which is a subdiffusive motion of the chain's center of mass, which is not predicted by the Rouse model. There are attempts by analytical theory [25, 26], but this issue is still under investigation. To some extent, the deviations might be trivially due to the fact that applicability of the Rouse model requires that the chains are long enough to satisfy a Gaussian description, and at the same time short enough that reptation does not yet play a role. Usually this window of chain lengths is very small, and in many cases it practically does not exist.

Let us now discuss the consequences of the Rouse model. It is interesting to note that for a Gaussian (RW) chain the model can be solved *exactly*. The reason is that in this case V is a harmonic potential, i. e. the equation of motion is linear. The problem is then mathematically very similar to phonons in a one–dimensional solid. Analogously to phonons one introduces the so–called Rouse modes

$$\boldsymbol{X}_p = \sqrt{2}\, N^{-1/2} \sum_{i=1}^{N} \boldsymbol{r}_i \cos\left[\frac{p\pi}{N}\,(i-1/2)\right], \qquad p = 1, \ldots N-1, \tag{15}$$

whose equations of motion decouple. Each mode is characterized by a mean square amplitude

$$\left\langle \boldsymbol{X}_p^2 \right\rangle = \frac{b^2}{4\sin^2\left(\frac{p\pi}{2N}\right)}, \tag{16}$$

and a relaxation function

$$\left\langle \boldsymbol{X}_p(t) \cdot \boldsymbol{X}_p(0) \right\rangle = \left\langle \boldsymbol{X}_p^2 \right\rangle \exp\left(-\frac{t}{\tau_p}\right), \tag{17}$$

where the mode relaxation time τ_p is given by

$$\tau_p^{-1} = \frac{12 k_B T}{\zeta b^2} \sin^2\left(\frac{p\pi}{2N}\right). \tag{18}$$

Note that for long chains and small mode number this is approximated by the scaling law $\tau_p \propto (N/p)^2$. The longest relaxation time is $\tau_1 \equiv \tau_R$, the so–called Rouse time, scaling as $\tau_R \propto N^2 \propto R^4$. At this point it is useful to introduce a dynamic exponent z, which relates the length scale R with the corresponding time scale τ_R via

$$\tau_R \propto R^z. \tag{19}$$

We therefore see $z = 4$ for the RW Rouse model.

The mean square displacement of a single monomer, $\left\langle \Delta \boldsymbol{r}^2 \right\rangle$, can be written exactly as a complicated expression, which however behaves asymptotically as $\left\langle \Delta \boldsymbol{r}^2 \right\rangle \propto t^{1/2}$ for times $\tau_m \ll t \ll \tau_R$, where τ_m is the microscopic time scale given by the time a monomer needs to move its own size, $\tau_m = b^2(\zeta/k_B T)$. The zeroth Rouse mode describes the motion of the center of mass, which is pure diffusion on all time scales, with diffusion constant

$$D = \frac{k_B T}{N\zeta}. \tag{20}$$

For times large compared to τ_R, *all* monomers move just diffusively with diffusion constant D.

It is physically more instructive to derive the results of the Rouse model directly by scaling reasoning, and to do this for RW and SAW statistics simultaneously. Looking at first at the equation of motion for the center of

mass, one notes that the drift (force) terms all cancel, due to Newton's third law. Furthermore, the friction coefficients of all the monomers simply add up; hence Eq. 20 is derived immediately. The scaling law is

$$D \propto N^{-1} \propto R^{-1/\nu}. \tag{21}$$

Furthermore, one estimates the longest relaxation time τ_R via the consideration that the object will just move its own size within τ_R:

$$D\tau_R \propto R^2 \qquad\qquad \tau_R \propto R^{2+1/\nu}, \tag{22}$$

from which we read off $z = 2 + 1/\nu$ for the general case. Furthermore, scaling tells us that the mean square displacement of the single monomer should follow a power law for times $\tau_m \ll t \ll \tau_R$, simply because the system has no further important time scale. Requiring $\left\langle \Delta r^2 \right\rangle \sim R^2$ for $t \sim \tau_R$ fixes the exponent as

$$\left\langle \Delta r^2 \right\rangle \propto t^{2/z}. \tag{23}$$

The physical picture is that originally a monomer can move freely (with diffusion constant $D_0 = k_B T/\zeta$), while at later times it has to drag more and more neighboring monomers along. Therefore the effective diffusion constant systematically decreases with time, until (at $t = \tau_R$) the whole chain is dragged along.

Finally, the single–chain dynamic structure factor, defined as

$$S(k,t) = \frac{1}{N} \left\langle \sum_{ij} \exp\left(i\boldsymbol{k} \cdot (\boldsymbol{r}_i(t) - \boldsymbol{r}_j(0))\right) \right\rangle, \tag{24}$$

satisfies the scaling relation

$$S(k,t) = k^{-1/\nu} f\left(k^2 t^{2/z}\right) \tag{25}$$

for $b \ll k^{-1} \ll R$ and $\tau_m \ll t \ll \tau_R$.

1.4 Zimm Model

As already indicated above, in dilute solutions it is necessary to take hydrodynamic momentum transport into account. The main effect is the so–called "hydrodynamic interaction": A monomer i is randomly kicked by its solvent surrounding, and is moved by a certain random displacement (per unit time) $\boldsymbol{\rho}_i$. Another monomer j suffers a displacement $\boldsymbol{\rho}_j$. Now, the motion of the solvent particles near $\boldsymbol{r}_i$ is *highly correlated* with that at position $\boldsymbol{r}_j$, due to fast diffusive momentum transport through the solvent. Strictly spoken, this correlation only occurs at some later time (the time which the "signal" needs to travel from $\boldsymbol{r}_i$ to $\boldsymbol{r}_j$), but this is usually quite short compared to the time which the monomers i and j need to travel considerably. As already discussed,

the momentum transport occurs with the "diffusion constant" η_{kin}, while the particles move (initially) with diffusion constant $D_0 = k_B T/\zeta$. The dimensionless ratio is called the Schmidt number $Sc = \eta_{kin}/D_0$; its value controls how accurate the neglect of retardation effects is. Typical numbers for Sc in dense fluids are of the order $Sc \approx 10^2$.

Thus, in contrast to the Rouse case, where the stochastic displacements exhibit no correlations between different particles and between different spatial directions, we now have a non–trivial correlation function $\langle \boldsymbol{\rho}_i(t) \otimes \boldsymbol{\rho}_j(t') \rangle \propto \delta(t-t')$, where the tensorial nature is due to the incompressibility constraint of the solvent flow, and the delta function in time expresses the neglect of retardation effects. (The symbol $\otimes$ denotes the tensor product.)

Furthermore, a force $\boldsymbol{F}$ acting on a monomer at $\boldsymbol{r}_i$ will generate a surrounding flow around it. Again neglecting retardation, one can use the *stationary* Stokes equation to calculate the resulting flow field. The solution is [1] the so–called Oseen tensor

$$\boldsymbol{u}(\boldsymbol{r}) = \overleftrightarrow{T}\,(\boldsymbol{r} - \boldsymbol{r}_i) \cdot \boldsymbol{F} \tag{26}$$

$$\overleftrightarrow{T}\,(\boldsymbol{r}) = \frac{1}{8\pi\eta r}(\overleftrightarrow{1} + \hat{r} \otimes \hat{r}), \tag{27}$$

where $\hat{r}$ is the unit vector in the direction of $\boldsymbol{r}$.

We now write down the most general Langevin equation which is still memory–free, does not use more variables than the monomer coordinates, and satisfies the fluctuation–dissipation theorem (to assure the correct equilibrium distribution function):

$$\frac{d}{dt}\boldsymbol{r}_i = \sum_j \overleftrightarrow{\mu}_{ij} \cdot \boldsymbol{F}_j + \boldsymbol{\rho}_i \tag{28}$$

where $\overleftrightarrow{\mu}_{ij}$ is the mobility tensor, and the stochastic displacements satisfy the relation

$$\langle \boldsymbol{\rho}_i(t) \otimes \boldsymbol{\rho}_j(t') \rangle = 2k_B T\, \overleftrightarrow{\mu}_{ij}\, \delta(t-t'). \tag{29}$$

We only consider mobility tensors which are divergence–free, and hence we need not worry about "spurious drift" terms [1]. Now, since the monomers are essentially just "embedded" in the surrounding flow, we can identify (at least approximately) $\overleftrightarrow{\mu}_{ij} = \overleftrightarrow{T}\,(\boldsymbol{r}_i - \boldsymbol{r}_j)$ (note that both objects just describe the velocity response to a force). This holds of course only for $i \neq j$; for the diagonal elements we assume the Rouse form $\overleftrightarrow{\mu}_{ii} = \overleftrightarrow{1}\,/\zeta$.

The scaling analysis of the Zimm model now proceeds along the same lines as for the Rouse model. First, we study the center–of–mass diffusion constant. In the short–time limit, it is easy to show that the center of mass moves with the Kirkwood diffusion constant

$$D^{(K)} = \frac{D_0}{N} + \frac{k_B T}{6\pi\eta} \left\langle \frac{1}{R_H} \right\rangle, \tag{30}$$

and this differs only marginally from the long–time value [27]. Since the second term strongly dominates in the long–chain limit, we find the scaling law

$$D \propto \frac{1}{R}, \tag{31}$$

indicating that the chain as a whole essentially moves like a Stokes sphere. The longest relaxation time (Zimm time τ_Z) is again found by requiring $D\tau_Z \sim R^2$, resulting in $\tau_Z \propto R^3$, or $z = 3$ independently from chain statistics. The dynamics is thus faster than in the Rouse case. All other scaling relations remain the same; one just has to use the appropriate values for ν and z.

1.5 Hydrodynamic Screening and Dynamic Crossover

Upon increasing the concentration, we have a static crossover from SAW to RW behavior; this is controlled by the blob size ξ. There is also a dynamic crossover from Zimm to Rouse behavior the physics of which had not fully been understood until very recently when a computer simulation [15] clarified the last remaining puzzles.

The underlying question is: How does the system get rid of its hydrodynamic correlations? Early attempts [28, 29] tried to attack this by studying the multiple scattering of the flow field. The attractive feature of such considerations is the fact that, under the assumption of a frozen polymer matrix, the analog of the Oseen tensor can be easily calculated: Assuming an array of fixed random obstacles with concentration c and friction coefficient ζ (per obstacle), the solvent flow field $\boldsymbol{u}$ experiences a friction force per unit volume of $-\zeta c\boldsymbol{u}$, such that the Stokes equation is modified to

$$\rho\frac{\partial}{\partial t}\boldsymbol{u} = \eta\nabla^2\boldsymbol{u} - \zeta c\boldsymbol{u}. \tag{32}$$

Its Green's function now exhibits a Debye–Hückel–like decay of the form $\propto (1/r)\exp(-r/\xi_H)$, where the hydrodynamic screening length ξ_H is found via $\eta\xi_H^{-2} = \zeta c$. The long–range Oseen decay is replaced by a short–range interaction, and thus Rouse behavior is expected on length scales beyond ξ_H, while Zimm behavior should apply on short length scales.

De Gennes [30] has critized this approach for the following reasons: (i) The chains are not at all fixed obstacles, but, on the contrary, enslaved to the surrounding flow and just dragged along; (ii) the predicted scaling $\xi_H \propto c^{-1/2}$ would imply that ξ and ξ_H are not proportional to each other, which makes a scaling analysis difficult if not impossible.

De Gennes' solution to the puzzle [30] is based on the following argument: A description in terms of fixed obstacles *is* justified, however only on length scales beyond the blob size ξ. For these larger length scales, it is the *entanglements* (not in the sense of reptation theory, but rather in the sense of *mutual interaction*) which cause a hindrance in polymer mobility compared to fast

Zimm motion. Since a Zimm chain behaves essentially like a Stokes sphere, one arrives at the picture of blobs "hooked up" in a temporary gel. Therefore the appropriate obstacles are not the monomers, but rather the blobs, with Stokes friction coefficient $\zeta_{blob} \sim \eta\xi$, and concentration $c_{blob} \sim \xi^{-3}$. Inserting these relations into $\eta\xi_H^{-2} = \zeta_{blob}c_{blob}$, one finds $\xi_H \sim \xi$, i. e. the length scales are (apart from prefactors) *identical*.

This has been confirmed by most experiments [31, 32]; however, the picture of clean Rouse motion beyond the length scale ξ was questioned by the observation of "incomplete screening" in the data of neutron spin echo experiments on labeled chains [33], where a clear Zimm–like contribution was found. The results of our recent simulation [15] revealed the solution of this puzzle: It is not sufficient to look at the problem just in terms of length scales, but one has to consider the time scales as well, and distinguish between the cases $t \ll \tau_\xi$ and $t \gg \tau_\xi$, where τ_ξ is the blob (Zimm) relaxation time, $\tau_\xi \sim \eta\xi^3/(k_BT)$. Since one has to wait (on average) for a time of order τ_ξ until an entanglement (or, synonymously: an interaction, a chain–chain collision) occurs, there is no screening whatsover for short times. The motion is rather free Zimm relaxation on *all* length scales, and the chains are just dragged along with the flow. After τ_ξ, the interactions are felt, and the blob screening mechanism sets in, resulting in Rouse–like motion. This, however, has only an effect on the length scales beyond ξ, since at that time all correlations within the blob have already decayed.

Let us now discuss the numerical results of Ref. [15]. In order to simulate a real semidilute solution, it is necessary to resolve both the RW regime at large length scales and the SAW regime within the blob. In order to observe random coil behavior in computer chains, a certain minimum number of monomers is necessary. According to our experience, one needs at least $N \approx 30$ monomers to clearly see the scaling behavior of either a RW or a SAW. For our semidilute solution, this means that we need roughly 30 monomers per blob, and roughly 30 blobs per chain. Therefore the minimum chain length is roughly $N = 1000$. For such a chain we then expect a mean size of $R = 30^\nu \times 30^{1/2} \approx 40$, in units of the bond length. In order to safely exclude self–overlaps, one would like to make the linear size of the simulation box (with periodic boundary conditions) substantially bigger. Our largest system therefore had linear box size $L = 88$. The concentration is then obtained via $\xi \approx 30^\nu \approx c^{-0.77}$ or $c \approx 0.066$. The total number of monomers in the $L = 88$ box then results as 45000. In the actual simulation, we studied 50 chains of length $N = 1000$ *which is essentially the smallest system to study a semidilute solution.* For such a system we calculated the single–chain dynamic structure factor $S(k, t)$. The static ($t = 0$) structure factor revealed the expected RW and SAW regimes. To analyze dynamic scaling, one plots the data as a function of the scaling argument $k^2t^{2/z}$. Indeed we found Zimm behavior ($z = 3$) at short times and small length scales, while the data show RW Rouse behavior ($z = 4$) for late times, large length scales. A particularly careful analysis was necessary to distinguish the short–time and long–time regimes (Zimm vs. Rouse) for

the data at large length scales ($k\xi < 1$). In order to enhance the short–time region, Fig. 2 shows $-\ln[S(k,t)/S(k,0)]$ instead of simply $S(k,t)/S(k,0)$ as a function of the scaling argument, for *both* Zimm and Rouse scaling. It is clearly seen that Zimm scaling applies for the short times $t \ll \tau_\xi$, while Rouse scaling holds for the later times $t \gg \tau_\xi$.

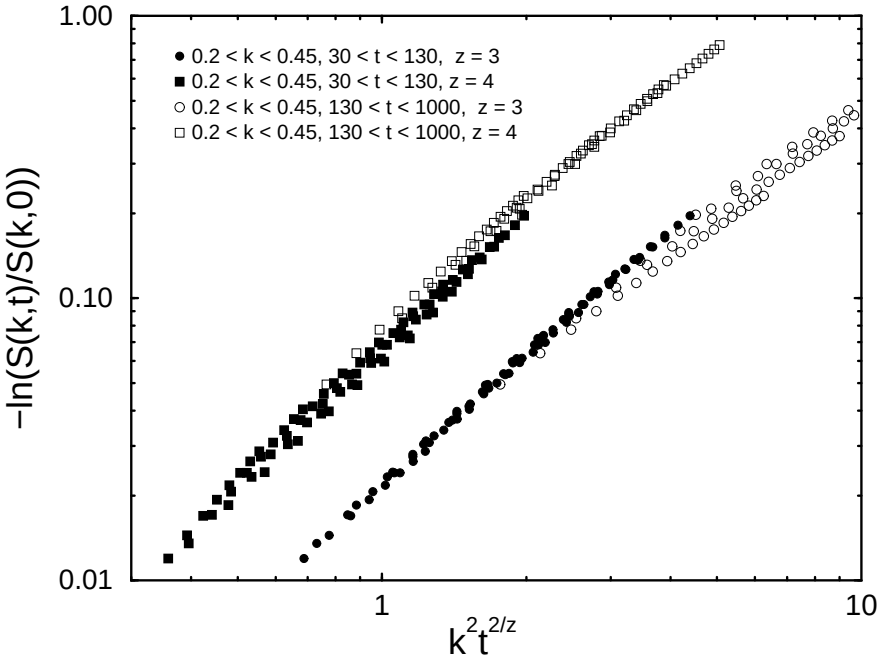

Fig. 2. Scaling plot of single–chain dynamic structure factor data, for both Rouse and Zimm scaling, taken from Ref. [15]. For more details, see the text and Ref. [15].

2 Simulation Methods

2.1 Overview

We now ask the question: What is the right way to simulate a system with hydrodynamic interactions? For simple problems, many methods will work, but for a challenging application like the system of Ref. [15] it is necessary to choose and design the method carefully.

The most straightforward approach would be Brownian dynamics, where just Eqn. 28 is simulated directly, either for a single–chain system, or a many–chain system. However, this will not work for a system of 50000 monomers.

Each time step one would have to calculate a 150000×150000 matrix, and to calculate its square root, in order to find the stochastic displacements. This is beyond the capacity of today's computers. The unfavorable scaling of the computational complexity with the number of Brownian particles makes the method only feasible for small systems. In Ref. [27] we studied the Zimm equation of motion for a single chain, and calculated the diffusion constant accurately. The longest chain which was accessible was $N = 200$.

It is therefore quite clear that one needs a method which scales *linearly* with the number of Brownian particles. An $O(N)$ algorithm for evaluating hydrodynamic interactions has indeed been developed, in close analogy to the fast multipole method for electrostatic interactions [34]. However, this is practically only applicable to deterministic problems (like sedimentation) where no thermal noise needs to be considered. For stochastic simulations, the problem of calculating the matrix square root remains.

Therefore the most promising route is to simulate the momentum transport through the solvent *explicitly* via some computational scheme. The most straightforward way to do this is of course Molecular Dynamics (MD), where the Brownian particles are immersed into a bath of solvent particles, and Newton's equations of motion are solved (without modification like a thermostat etc.). However, this leads to an unnecessarily large computational effort. One needs to follow the motion of each solvent particle down to the time scale of the local oscillation of the particles in their cages, while they have essentially no function except for transporting momentum. One rather would like to simulate the solvent on a somewhat larger time scale, in order to save computer time. Essentially, this is coarse–graining with respect to time scales (not so much with respect to length scales, since one would not like to lose resolution in representing the hydrodynamic interactions).

Indeed, there are several ways to do this. One approach, which is conceptually particularly close to MD, is Dissipative Particle Dynamics (DPD) [35–46], which has become a quite popular method for "mesoscopic" simulations of the dynamics of soft–matter systems. One makes the particles quite soft, in order to afford a large time step, and also adds a momentum–conserving Langevin thermostat. It should be stressed that these two components are conceptually completely independent, and can also be implemented independently. It is therefore relatively straightforward to change an existing MD code into a DPD simulation, by just adding the thermostat. More details on DPD will follow below.

Another simulation aspect which needs appreciation is the issue of equilibration. This is of course completely uninteresting for nonequilibrium studies, which become more and more important, but for studying the dynamics in strict thermal equilibrium this is of paramount importance. Soft matter objects with internal degrees of freedom (like polymer chains, but also membranes) tend to have complex configuration spaces and large relaxation times. On the other hand, one often is not interested in following the dynamic correlation functions all the way up to the longest relaxation time. Such a situation

is exactly present in the study of Ref. [15], where only the dynamics up to τ_ξ (and somewhat beyond) is needed, but not up to τ_R. One would therefore like to be able to equilibrate the system with a fast Monte Carlo algorithm, in order to shortcut the slow physical dynamics, and to use the generated configurations for starting runs with realistic (slow) dynamics, over which one averages. For dilute systems, the ideal way to do this is to *completely disregard the solvent in the equilibration procedure*. However, this requires the solvent to be *structureless*. A structured solvent (i. e. particles with some non–trivial interaction potential) always modifies the potential of mean force of the solute, and this is not known in advance. Therefore only the coupled solute–solvent system can be equilibrated "cleanly", i. e. without introducing systematic errors. Conversely, for a structureless solvent the potential of mean force of the solute is identical to the "bare" potential (i. e. the interaction without solvent). For a particle method, this means that the solvent should be an ideal gas. In this case, however, MD is not applicable since all particle trajectories are trivial, and there are no collisions. Conversely, DPD is able to simulate an ideal gas with realistic dynamics, because collisions are effectively implemented via the thermostat. Another particle method, which is also based on ideal gas particles, is multi–particle collision dynamics (MPCD) [47], where collisions are implemented via local stochastic updating rules which conserve energy and momentum. A brief outline of this approach is included below.

Yet another approach to simulate momentum transport through the solvent is to solve the (Navier–) Stokes equation (in a deterministic or stochastic version) on a grid, and to couple an MD system for the solute to such a simulation. This yields a structureless solvent automatically. Solving the hydrodynamics can be done either by a finite–difference scheme, or by the lattice Boltzmann method (LBM) [48]. The latter has become also quite popular for soft matter systems, in particular colloidal suspensions [49–57], and will be discussed below. Compared to DPD, it has the disadvantage that the underlying theory is slightly involved, and that the coupling to the solute system (which is still simulated by some MD–like algorithm) is not a straightforward consequence of the method, but rather must be constructed by hand. The advantage, however, is that it is based on a tight data structure, with the consequence that it is computationally quite efficient, rather straightforward to implement, and ideally suited for parallel computers (the only communication is just the sending of data in the streaming step, while the collision step needs only local data). Another important advantage of a grid–based method is that thermal fluctuations may be both turned on (necessary for Brownian motion, for instance), or off (for some nonequilibrium studies like sedimentation thermal fluctuations are not needed, in particular if the solute system does not have fluctuating internal degrees of freedom). This flexibility is a quite useful aspect, and not present in particle methods, which *always* exhibit thermal fluctuations. A noise–free simulation is of course much cheaper than a noisy one, since no cumbersome averaging is necessary (except, perhaps, over initial conditions).

2.2 Dissipative Particle Dynamics (DPD)

Dissipative Particle Dynamics is essentially MD, where a momentum–conserving Langevin thermostat is added. The method is best understood by contrasting it to the older method of Stochastic Dynamics (SD) [58], which also adds a Langevin thermostat to MD, but does *not* conserve the momentum. The formal development is most transparent if we start from Hamilton's formulation of Newton's equations of motion:

$$\frac{d}{dt}q_i = \frac{\partial \mathcal{H}}{\partial p_i} \tag{33}$$

$$\frac{d}{dt}p_i = -\frac{\partial \mathcal{H}}{\partial q_i}, \tag{34}$$

where the q_i denote the generalized coordinates, and the p_i the generalized canonically conjugate momenta, while $\mathcal{H}$ is the Hamiltonian of the system. Adding friction and noise, we obtain the SD equations of motion:

$$\frac{d}{dt}q_i = \frac{\partial \mathcal{H}}{\partial p_i} \tag{35}$$

$$\frac{d}{dt}p_i = -\frac{\partial \mathcal{H}}{\partial q_i} - \zeta_i \frac{\partial \mathcal{H}}{\partial p_i} + \sigma_i f_i; \tag{36}$$

here ζ_i is the friction coefficient for the ith degree of freedom (note that $\partial \mathcal{H}/\partial p_i$, for usual Cartesian coordinates, is nothing but the velocity), σ_i denotes the noise strength, while $\langle f_i \rangle = 0$ and $\langle f_i(t) f_j(t') \rangle = 2\delta_{ij}\delta(t - t')$. We can even allow that the friction constants ζ_i and the noise strengths σ_i depend on the coordinates q_i (but not on the momenta p_i).

We now switch to an equivalent description of the stochastic process, where we study the time evolution of the probability density in phase space, $P(\{q_i\}, \{p_i\}, t)$. This is quite analogous to switching from Hamilton's equations of motion to the Liouville equation in classical mechanics. The equation of motion for P is called the Fokker–Planck equation. Its shape can be derived directly from the Langevin equation, using a standard procedure described in textbooks on stochastic processes (see, e. g. Refs. [59, 60] or [61]). For the present case, one obtains

$$\frac{\partial}{\partial t}P = \mathcal{L}P, \tag{37}$$

where $\mathcal{L}$ is the Fokker–Planck operator, which is naturally decomposed into two parts,

$$\mathcal{L} = \mathcal{L}_H + \mathcal{L}_{SD}, \tag{38}$$

where the first part refers to the Hamiltonian part of the dynamics (it is nothing but the Liouville operator),

$$\mathcal{L}_H = -\sum_i \frac{\partial}{\partial q_i}\frac{\partial \mathcal{H}}{\partial p_i} + \sum_i \frac{\partial}{\partial p_i}\frac{\partial \mathcal{H}}{\partial q_i}$$

$$= -\sum_i \frac{\partial \mathcal{H}}{\partial p_i}\frac{\partial}{\partial q_i} + \sum_i \frac{\partial \mathcal{H}}{\partial q_i}\frac{\partial}{\partial p_i}, \tag{39}$$

while the second part is due to friction and noise,

$$\mathcal{L}_{SD} = \sum_i \frac{\partial}{\partial p_i}\left[\zeta_i \frac{\partial \mathcal{H}}{\partial p_i} + \sigma_i^2 \frac{\partial}{\partial p_i}\right]. \tag{40}$$

In order to describe a system in thermal equilibrium, the Boltzmann distribution must be the stationary solution of the Fokker–Planck equation:

$$\mathcal{L} \exp\left(-\beta \mathcal{H}\right) = 0, \tag{41}$$

where $\beta = 1/(k_B T)$. For the Hamiltonian part, this relation is identically fulfilled. Therefore, the condition results in

$$\sum_i \frac{\partial}{\partial p_i}\left[\zeta_i \frac{\partial \mathcal{H}}{\partial p_i} - \beta \sigma_i^2 \frac{\partial \mathcal{H}}{\partial p_i}\right] \exp\left(-\beta \mathcal{H}\right) = 0. \tag{42}$$

Hence the relation

$$\sigma_i^2 = k_B T \zeta_i \tag{43}$$

must hold. This is the fluctuation–dissipation theorem (FDT), i. e. the temperature is a result of the balance between friction and noise strength. However, the momentum is *not conserved*, as one can check immediately from the equations of motion. Rather, the center of mass of the system diffuses. The algorithm also violates Galilean invariance, since it dampens the *absolute* velocities, thus labeling the "laboratory frame" as special, which is of course unphysical. These are the reasons why SD is useless for hydrodynamic simulations. It can be shown [61, 62] that this unphysical behavior can be expressed in terms of a hydrodynamic screening length $\xi = [\eta/(n\zeta)]^{1/2}$. Here, we have assumed a constant friction, while n is the particle number density. The arguments to derive this are essentially the same as those presented in Sec. 1.5 for a frozen matrix of frictional obstacles.

Dissipative Particle Dynamics (DPD) has been developed to cure this problem, and to simulate hydrodynamic phenomena in fluids on a mesoscopic scale. DPD, as it is usually described in the literature, consists of two parts: (i) Introduction of very soft interparticle potentials in order to facilitate a large time step, and (ii) introduction of a Galilei invariant thermostat, which is similar to SD, but dampens *relative* velocities, and applies the stochastic kicks to *pairs* of particles such that Newton's third law (i. e. momentum conservation) is satisfied. As the procedure is also completely local, it is therefore suitable for the description of (isothermal) hydrodynamics. Unfortunately, it is often not made sufficiently clear that these two parts are *completely unrelated*, i. e. that one can use the DPD thermostat with "conventional" hard potentials, and that one can go from a working MD code to DPD, just as one would

go to SD. A technical problem of typical DPD simulations is the fact that, due to the soft potentials, they are run with extremely large time steps. This results in unacceptably large discretization errors. Currently this problem is under thorough investigation [41–46]. We will from now on exclusively focus on the thermostat aspect of DPD. As Espanol and Warren [37] have shown, the structure of the FDT for DPD is very similar to the SD case. A particularly useful application of the DPD thermostat, which is just presently being appreciated, is its use in *nonequilibrium* studies like the simulation of steady–state Couette flow. Nonequilibrium steady states are characterized by a constant nonzero rate of entropy production, usually showing up as viscous heat. This produced entropy must be removed from the system, and therefore such simulations are usually coupled to a thermostat (an alternative approach, which rather removes the entropy by a Maxwell demon, has recently been developed by Müller–Plathe [63, 64]). Before the advent of DPD, it was a non–trivial problem to introduce the thermostat in such a way that it would not prefer a certain profile (so–called "profile–unbiased thermostats", see Ref. [65]). The DPD thermostat solves this problem in a very natural and straightforward way [66].

In practice, DPD simulations are done as follows: We first define two functions, $\zeta(r)$, the relative friction coefficient for particle pairs with interparticle distance r, and $\sigma(r)$, the noise strength for a stochastic kick applied to the same particle pair. We will show below that the FDT implies the relation

$$\sigma^2(r) = k_B T \zeta(r), \tag{44}$$

in close analogy to SD. The function has a finite range, such that only near neighbors are taken into account.

Defining $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j = r_{ij}\hat{r}_{ij}$, we then obtain the friction force on particle i by projecting the relative velocities on the interparticle axes:

$$\boldsymbol{F}_i^{(fr)} = -\sum_j \zeta(r_{ij}) \left[(\boldsymbol{v}_i - \boldsymbol{v}_j) \cdot \hat{r}_{ij}\right] \hat{r}_{ij}; \tag{45}$$

it is easy to see that the relation $\sum_i \boldsymbol{F}_i^{(fr)} = 0$ holds. Similarly, we get the stochastic forces along the interparticle axes:

$$\boldsymbol{F}_i^{(st)} = \sum_j \sigma(r_{ij})\, \eta_{ij}(t)\, \hat{r}_{ij}, \tag{46}$$

where the noise η_{ij} satisfies the relations $\eta_{ij} = \eta_{ji}$, $\langle \eta_{ij} \rangle = 0$, and

$$\langle \eta_{ij}(t) \eta_{kl}(t') \rangle = 2(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})\delta(t - t'), \tag{47}$$

such that different pairs are statistically independent. As before, one easily shows $\sum_i \boldsymbol{F}_i^{(st)} = 0$. The equations of motion,

$$\frac{d}{dt}\boldsymbol{r}_i = \frac{1}{m_i}\boldsymbol{p}_i, \tag{48}$$

$$\frac{d}{dt}\boldsymbol{p}_i = \boldsymbol{F}_i + \boldsymbol{F}_i^{(fr)} + \boldsymbol{F}_i^{(st)}, \tag{49}$$

where m_i is the mass of the ith particle, and $\boldsymbol{p}_i$ its momentum, therefore indeed conserve the total momentum, as the conservative forces $\boldsymbol{F}_i$ satisfy Newton's third law. The Fokker–Planck operator can then be written as

$$\mathcal{L} = \mathcal{L}_H + \mathcal{L}_{DPD}, \tag{50}$$

where $\mathcal{L}_H$ again describes the Hamiltonian part with $\mathcal{L}_H \exp(-\beta\mathcal{H}) = 0$ (cf. Eq. 39), and $\mathcal{L}_{DPD}$ is given by

$$\begin{aligned}
\mathcal{L}_{DPD} &= \sum_{ij} \zeta(r_{ij})\hat{r}_{ij} \cdot \frac{\partial}{\partial \boldsymbol{p}_i}\left[\hat{r}_{ij} \cdot \left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{p}_i} - \frac{\partial \mathcal{H}}{\partial \boldsymbol{p}_j}\right)\right] \\
&- \sum_{i\neq j} \sigma^2(r_{ij}) \left(\hat{r}_{ij} \cdot \frac{\partial}{\partial \boldsymbol{p}_i}\right)\left(\hat{r}_{ij} \cdot \frac{\partial}{\partial \boldsymbol{p}_j}\right) \\
&+ \sum_i \sum_{j(\neq i)} \sigma^2(r_{ij}) \left(\hat{r}_{ij} \cdot \frac{\partial}{\partial \boldsymbol{p}_i}\right)^2 \\
&= \sum_i \sum_{j(\neq i)} \hat{r}_{ij} \cdot \frac{\partial}{\partial \boldsymbol{p}_i}\Bigg[\zeta(r_{ij})\hat{r}_{ij} \cdot \left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{p}_i} - \frac{\partial \mathcal{H}}{\partial \boldsymbol{p}_j}\right) \\
&+ \sigma^2(r_{ij})\hat{r}_{ij} \cdot \left(\frac{\partial}{\partial \boldsymbol{p}_i} - \frac{\partial}{\partial \boldsymbol{p}_j}\right)\Bigg].
\end{aligned} \tag{51}$$

In the stochastic term, we have first taken into account the off–diagonal terms (cross–correlations, which are actually anti–correlations between the neighbors). The prefactors for the diagonal terms are given by the sum of all the mean square noise strengths from all the neighbors. Applying this operator to $\exp(-\beta\mathcal{H})$, we find that the FDT is satisfied if $\sigma^2(r) = k_B T \zeta(r)$.

2.3 Multi–Particle Collision Dynamics (MPCD)

As already mentioned, DPD simulations can be run for the special case of vanishing interaction potential, i. e. in the ideal gas limit. If we discretize this procedure in terms of, say, the Verlet algorithm, we arrive at a method where free particle propagation (i. e. update in real space without update in momentum space) alternates with "collisions" (update in momentum space without update in real space). Due to their dissipative nature, these DPD collisions conserve momentum but not the energy.

Malevanets and Kapral [47] have introduced a method which is also based on collisions of ideal gas particles. However, the collisions are now implemented

by a simple Monte Carlo procedure such that *both* momentum and energy are conserved. Starting from a set of particle coordinates $\boldsymbol{r}_i$ and a set of particle velocities $\boldsymbol{v}_i$, $i = 1, \ldots, N$, one first performs a streaming step (free propagation by a time step h)

$$\boldsymbol{r}_i(t+h) = \boldsymbol{r}_i(t) + h\boldsymbol{v}_i(t). \tag{52}$$

This is followed by a collision step, which is facilitated by sub–dividing the simulation box into sub–boxes. For each sub–box, one determines the set of particles residing in it. For one particular sub–box, let these particles be enumerated by $i = 1, \ldots, n$. These "collide" with each other by the following procedure:

- Determine the local center–of–mass velocity:

$$\boldsymbol{v}_{CM} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}_i. \tag{53}$$

- For each particle in the sub–box, perform a Galileo transformation into the local center–of–mass system:

$$\tilde{\boldsymbol{v}}_i = \boldsymbol{v}_i - \boldsymbol{v}_{CM}. \tag{54}$$

- Within the local center–of–mass system, rotate all velocities within the sub–box by a random rotation matrix $\overleftrightarrow{R}$:

$$\tilde{\boldsymbol{v}}_i{}' = \overleftrightarrow{R}\, \tilde{\boldsymbol{v}}_i. \tag{55}$$

- Transform back into the "laboratory" system:

$$\boldsymbol{v}_i' = \tilde{\boldsymbol{v}}_i{}' + \boldsymbol{v}_{CM}. \tag{56}$$

One sees immediately that this procedure satisfies locality as well as the conservation of mass, momentum, and energy. Ihle and Kroll [67] have pointed out that it is necessary to randomly shift the sub–boxes in order to avoid spurious effects and to restore full Galileo invariance. Furthermore, the fact that the dynamics is so simple makes it possible to derive analytic expressions for transport coefficients [68–70].

2.4 Lattice Boltzmann (LB)

The lattice Boltzmann method (LBM) works quite differently. Essentially, the method is the simulation of a fully discretized version of the (linearized) Boltzmann equation known from the kinetic theory of gases. One starts from a regular lattice (usually a simple–cubic lattice) with lattice spacing a; $\boldsymbol{r}$ denotes its sites. Furthermore, we introduce a finite (small) set of (dimensionless) vectors $\boldsymbol{c}_i$, such that $a\boldsymbol{c}_i$ is a vector connecting two sites on the lattice. The

set should be consistent with the point symmetry of the lattice. For example, on the simple–cubic lattice one would have six vectors $\boldsymbol{c}_i$ connecting to the nearest neighbors, and another twelve vectors to the next–nearest neighbors. Time is discretized in terms of a time step h, and the model allows only for a finite set of velocities. These are the vectors $(a/h)\boldsymbol{c}_i$. An object residing on a certain lattice site $\boldsymbol{r}$, and having the velocity $(a/h)\boldsymbol{c}_i$, would thus be moved to site $\boldsymbol{r} + a\boldsymbol{c}_i$ within one time step. A commonly used model is the 18–velocity model, where the vectors $\boldsymbol{c}_i$ correspond to the nearest and next–nearest neighbors. Sometimes an additional velocity $\boldsymbol{c}_i = 0$ is included (19–velocity model); this is however not necessary for simulating incompressible flow. The algorithm now works with real–valued variables $n_i(\boldsymbol{r}, t)$, denoting the "number of particles" which reside on site $\boldsymbol{r}$ at time t and have the velocity $(a/h)\boldsymbol{c}_i$. Denoting the particle mass with m, we find for the mass density at site $\boldsymbol{r}$ at time t

$$\rho(\boldsymbol{r}, t) = \frac{m}{a^3} \sum_i n_i(\boldsymbol{r}, t) \tag{57}$$

and for the momentum density

$$\boldsymbol{j}(\boldsymbol{r}, t) = \frac{m}{a^2 h} \sum_i n_i(\boldsymbol{r}, t) \boldsymbol{c}_i. \tag{58}$$

We can also introduce the streaming velocity $\boldsymbol{u}$ at site $\boldsymbol{r}$ via $\boldsymbol{u} = \boldsymbol{j}/\rho$. It should be noted that in many descriptions of the method the parameters m, a and h are set to unity, thus defining the unit system of the method. However, when coupling the LBM to an MD system, the latter has its own unit system. We prefer to use a unit system built upon MD, and for this purpose we need to keep the parameters. Furthermore, it should be noted that we do not consider the energy density. In this lecture, we only consider LBMs with mass and momentum conservation, while energy conservation (heat conduction etc.) is *not* taken into account. LBMs with proper inclusion of the energy have been developed [71], but are more complicated.

Now, the algorithm proceeds via the following steps:

1. Starting from the variables n_i, one calculates the hydrodynamic variables ρ and $\boldsymbol{j}$ on each lattice site.
2. From ρ and $\boldsymbol{j}$, one calculates a local pseudo–equilibrium distribution n_i^{eq}. It should be stressed that this is done for each site separately. Since the variables ρ and $\boldsymbol{j}$ differ from site to site, one has a different distribution n_i^{eq} on each site. The kinetic–theory analogue would be a Maxwell–Boltzmann velocity distribution centered around the hydrodynamic streaming velocity at position $\boldsymbol{r}$. Since n_i^{eq} and n_i correspond to the *same* hydrodynamic variables, we have $\sum_i n_i = \sum_i n_i^{eq}$ and $\sum_i n_i \boldsymbol{c}_i = \sum_i n_i^{eq} \boldsymbol{c}_i$ at each site.
3. Relaxation ("collisions"): The velocity distribution on the site is rearranged in order to bring it closer to the local equilibrium of that site. This is done via a linear process:

$$n_i \to n_i + \sum_j L_{ij}(n_j - n_j^{eq}). \tag{59}$$

In many cases, the matrix L_{ij} is just a multiple of the unit matrix. These are the so–called "lattice BGK" (Bhatnagar–Gross–Krook) methods. However, working with a nontrivial matrix causes no practical difficulties and allows one to get rid of non–hydrodynamic modes quickly [50]. In order to ensure mass and momentum conservation, the matrix should satisfy the conditions $\sum_i L_{ij} = 0$ and $\sum_i \boldsymbol{c}_i L_{ij} = 0$.

4. Streaming: The populations are displaced to new sites according to their velocities:

$$n_i(\boldsymbol{r}, t) \to n_i(\boldsymbol{r} + a\boldsymbol{c}_i, t + h). \tag{60}$$

This is the only step which is not completely local.

Further specification of the algorithm requires to give prescriptions for the calculation of n_i^{eq}, and of the relaxation procedure. A common procedure is to use the polynomial ansatz [50]

$$n_i^{eq} = \frac{a^3}{m}\rho\left(A_i + B_i \boldsymbol{c}_i \cdot \boldsymbol{u}\frac{h}{a} + C_i u^2 \frac{h^2}{a^2} + D_i(\boldsymbol{c}_i \cdot \boldsymbol{u})^2 \frac{h^2}{a^2}\right), \tag{61}$$

where symmetry requires that A_i should only depend on the neighbor shell, but not on the direction within it, and the same holds also for B_i, C_i, D_i. The 18–velocity model thus has eight coefficients. These are determined via the following requirements:

- n_i^{eq} should produce the correct hydrodynamic variables ρ and $\boldsymbol{j}$, as mentioned above.
- The stress tensor constructed from n_i^{eq},

$$\overleftrightarrow{\Pi}^{eq} = \frac{m}{a^3}\frac{a^2}{h^2}\sum_i n_i^{eq} \boldsymbol{c}_i \otimes \boldsymbol{c}_i, \tag{62}$$

should have the hydrodynamic form

$$\overleftrightarrow{\Pi}^{eq} = \rho c_s^2 \overleftrightarrow{1} + \rho \boldsymbol{u} \otimes \boldsymbol{u}; \tag{63}$$

here we have assumed the equation of state of an ideal gas with sound velocity c_s (other equations of state can be implemented [72]).

- The viscosity tensor (which, on a cubic lattice, will in general be a fourth–rank tensor with cubic anisotropy) should exhibit the full rotational symmetry, such that there are only shear and bulk viscosity. This is the main reason why nearest and next–nearest neighbor shells are used: The coefficients can be adjusted in such a way that the anisotropic contributions from the two shells just cancel.
- For $\boldsymbol{u} = 0$ both shells should contain the same number of particles. This is useful for numerical stability [50].

These conditions suffice to determine the coefficients and the parameter c_s uniquely.

The relaxation operator is determined via the following considerations: Apart from mass and momentum conservation, which already give four conditions, one observes that the linear relaxation of n_i towards n_i^{eq} corresponds to a linear relaxation of the stress tensor

$$\overleftrightarrow{\Pi} = \frac{m}{a^3}\frac{a^2}{h^2}\sum_i n_i \boldsymbol{c}_i \otimes \boldsymbol{c}_i \tag{64}$$

towards $\overleftrightarrow{\Pi}^{eq}$. We now require that this process exhibits two relaxation rates, one for the trace of $\overleftrightarrow{\Pi}$, corresponding to the bulk viscosity, and one for the trace–free part, corresponding to the shear viscosity. These parameters can thus be freely adjusted. Finally, we require [50] that the higher–order moments (non–hydrodynamic modes) are immediately removed after the relaxation process (this corresponds to eigenvalues -1). Under these circumstances, it turns out that the calculation of the new population (after the relaxation step) does not even require the implementation of the L_{ij} matrix. One rather has to simply update the pressure tensor, using the prescribed rates, and to use that result to calculate the new populations (again, the coefficients $A_i, \ldots, D_i$ are used) [50]. Via a Chapman–Enskog expansion one can show that this procedure yields hydrodynamic behavior in the macroscopic limit [50] *if the flow is incompressible, and the flow velocity is small compared to the sound velocity.* One particular advantage of the formulation based on the stress tensor is that the inclusion of thermal noise is quite straightforward: According to linear fluctuating hydrodynamics [73], the noise term occurs in the stress tensor, and therefore it can be directly added in the simulation code. For further details, see the original literature [50]. Recently, Ladd's procedure to include thermal noise has been extended, in order to produce better results for smaller length and time scales [74].

When coupling this to a system of Brownian particles, one can use two methods: The original approach by Ladd [50, 51] for colloidal suspensions was to use extended particles with a surface, and to implement a bounce–back rule to simulate the modification of the flow, plus the momentum transfer onto the particle. Combined with a lubrication correction for suspensions at high densities, this approach has produced excellent results for suspensions with hydrodynamic interactions [53].

For polymer solutions, we found a point–particle approach [75, 8] simpler and more efficient: While the solvent is run via the stochastic version of the LBM, the polymer system is simulated by MD augmented with friction and noise as in SD. However, the friction force is not $-\zeta \boldsymbol{v}$ ($\boldsymbol{v}$ particle velocity), but rather $-\zeta(\boldsymbol{v} - \boldsymbol{u})$, where $\boldsymbol{u}$ is the flow velocity at the position of the monomer, obtained via linear interpolation from the surrounding lattice sites. This determines the momentum transfer onto the particle which has come from the solvent. Momentum conservation requires that this momentum is

subtracted from the fluid. Details of this latter subtraction are not important; we used a procedure where we distributed the momentum transfer onto the surrounding sites using the same weights as the initial interpolation procedure. On each site, we then updated the n_i by requiring that the distance to n_i^{eq} remained unchanged. It can be shown that the coupled system does satisfy the FDT. Since locality, mass conservation, and momentum conservation are fulfilled, this procedure simulates hydrodynamic interactions faithfully, while being roughly 20 times faster than the analogous MD system with hard solvent particles. The lattice spacing was set roughly equal to the bond length; this is necessary to resolve the hydrodynamic interaction down to the relevant scales. We have recently shown [76, 77] that this approach can also be used to simulate colloidal particles, which are modeled as an arrangement of force centers like a "raspberry".

The friction coefficient ζ should be called "bare" friction coefficient, since the long–time single–particle mobility differs from $1/\zeta$ as a result of the long time tail. The correction can be quite strong, and actually depends on the lattice spacing a. This can be shown by the following consideration: We drag a particle with constant average velocity $\boldsymbol{v}$ and constant average force $\boldsymbol{F}$ through a fluid globally at rest. Our simulation procedure tells us that the force should be $\boldsymbol{F} = \zeta(\boldsymbol{v} - \boldsymbol{u})$, $\boldsymbol{u}$ being the flow velocity on the surrounding lattice sites, which are, on average, a distance of order a away. The Oseen tensor, in turn, tells us that u should be of order $u \sim F/(\eta a)$ or $u = F/(g\eta a)$, where g is some numerical coefficient. Combining these equations, we find for the mobility

$$\mu = \frac{1}{\zeta} + \frac{1}{g\eta a}; \tag{65}$$

this relation has been checked numerically [8]. The lattice thus provides a Stokes–like contribution to the mobility. It thus not only discretizes the hydrodynamics, but also regularizes it, i. e. it naturally cures the pathology that a point particle does not exist (note that in the continuum limit $a \to 0$ one would obtain an infinite mobility!). Since a is just a discretization parameter, the only conclusion is that ζ does not have any physical meaning. Rather, for comparing with experiments one should look at the "dressed" mobility μ.

3 Some Final Remarks

Although the presented material is highly selective, strongly reflecting my own research, I hope the present lecture has given a slight glimpse at the problems one encounters when simulating systems with hydrodynamic interactions, and also at the strategies which have been developed to cope with them. The development of so–called "mesoscopic" simulation methods ("somewhere between Molecular Dynamics and computational fluid dynamics (CFD)") for soft matter systems, with emphasis on hydrodynamics, is a quite active field of current research, and far from being closed. At the same time, applications are already

quite broad and continue to grow. Typical systems are polymers, colloids, liquid crystals, membranes, multiphase flows (e. g. spinodal decomposition of binary mixtures), electro–rheological systems, microfluidic devices, and biophysical systems, like, e. g., models for swimming bacteria. Furthermore, these methods (in particular the LBM) enter more and more the field of classical CFD, with applications like fluid turbulence or automotive engineering.

When asked about a judgmental statement about the presented methods, I feel very reluctant to give a recommendation. Reducing the various approaches to their bare essentials, it turns out that they all are not fundamentally different from each other. Although detailed benchmark comparisons have, to my knowledge, so far never been done, it is hard to conceive that the methods should differ very strongly in their computational performance. I personally like lattice methods, because they are easy to parallelize, and because thermal fluctuations can be turned on and off. However, for many interesting applications one may well get away with just a single processor, and for many soft matter systems, the inclusion of thermal fluctuations is anyways needed. Therefore, my personal opinion is that the choice of the method is, to a large extent, a matter of taste. The only recommendation which I can give is to try to understand as much physics as possible before running the simulation, and then ask the questions: (i) Which conservation laws (mass, momentum, energy) are needed to describe the physics correctly? (ii) Is the inclusion of thermal fluctuations really needed? (iii) Are large systems on a parallel machine needed? (iv) How much (molecular) structure of the fluid is needed? Starting from the answers to these questions, one should then be able to pick a suitable method and to construct a useful computational model.

Acknowledgments

The present paper is a corrected and extended version of a proceedings article which I wrote on occasion of the spring school *Computational Soft Matter: From Synthetic Polymers to Proteins*, organized by the Neumann Institute for Computing, Jülich (NIC) in 2004. I cordially thank the publisher (NIC) of the proceedings volume (same title, editors: N. Attig, K. Binder, H. Grubmüller, K. Kremer, ISBN 3–00–012641–4) for the kind permission to reuse this material.

Furthermore, I thank Patrick Ahlrichs, Ralf Everaers, Bo Liu, Igor Pasichnyk, Vladimir Lobaskin, Kurt Kremer, and Thomas Soddemann for a fruitful collaboration on the issues covered in the text, and Tony Ladd for stimulating discussions on mesoscopic simulation methods.

Last not least it is my pleasure to thank Christine Peter–Tittelbach, plus two further anonymous referees, for a careful and critical reading of the manuscript and useful suggestions.

References

1. Doi, M., Edwards, S.F.: The Theory of Polymer Dynamics. Oxford University Press, Oxford (1986)
2. Rouse, P.E. J. Chem. Phys. **21** (1953) 1272
3. Zimm, B.H. J. Chem. Phys. **24** (1956) 269
4. Pierleoni, C., Ryckaert, J.P. J. Chem. Phys. **96** (1992) 8539
5. Smith, W., Rapaport, D.C. Mol. Sim. **9** (1992) 25
6. Dünweg, B., Kremer, K. J. Chem. Phys. **99** (1993) 6983
7. Schlijper, A.G., Hoogerbrugge, P.J., Manke, C.W. J. Rheol. **39** (1995) 567
8. Ahlrichs, P., Dünweg, B. J. Chem. Phys. **111** (1999) 8225
9. Spenley, N.A. Europhys. Lett. **49** (2000) 534
10. Malevanets, A., Yeomans, J.M. Europhys. Lett. **52** (2000) 231
11. Kremer, K., Grest, G.S. J. Chem. Phys. **92** (1990) 5057
12. Paul, W., Binder, K., Heermann, D.W., Kremer, K. J. Phys. II (Paris) **1** (1991) 37
13. Paul, W., Binder, K., Heermann, D.W., Kremer, K. J. Chem. Phys. **95** (1991) 7726
14. Pütz, M., Kremer, K., Grest, G.S. Europhys. Lett. **49** (2000) 735
15. Ahlrichs, P., Everaers, R., Dünweg, B. Phys. Rev. E **64** (2001) 040501 (R)
16. Chang, R., Yethiraj, A. J. Chem. Phys. **114** (2001) 7688
17. Abrams, C.F., Lee, N.K., Obukhov, S.P. Europhys. Lett. **59** (2002) 391
18. Kikuchi, N., Gent, A., Yeomans, J.M. Europ. Phys. J. A **9** (2002) 63
19. de Gennes, P.G.: Scaling Concepts in Polymer Physics. Cornell University Press, Ithaca (1979)
20. Grosberg, A.Y., Khokhlov, A.R.: Statistical Physics of Macromolecules. AIP Press, New York (1994)
21. Alder, B., Wainwright, T. Phys. Rev. A **1** (1970) 18
22. Kopf, A., Dünweg, B., Paul, W. J. Chem. Phys. **107** (1997) 6945
23. Paul, W., Smith, G.D., Yoon, D.Y., Farago, B., Rathgeber, S., Zirkel, A., Willner, L., Richter, D. Phys. Rev. Lett. **80** (1998) 2346
24. Paul, W. Chemical Physics **284** (2002) 59
25. Rostiashvili, V.G., Rehkopf, M., Vilgis, T.A. Europ. Phys. J. B **6** (1998) 497
26. Guenza, M. Phys. Rev. Lett. **88** (2002) 025901
27. Liu, B., Dünweg, B. J. Chem. Phys. **118** (2003) 8061
28. Edwards, S.F., Freed, K.F. J. Chem. Phys. **61** (1974) 1189
29. Freed, K.F., Edwards, S.F. J. Chem. Phys. **61** (1974) 3626
30. de Gennes, P.G. Macromolecules **9** (1976) 594
31. Wiltzius, P., Haller, H.R., Carnell, D.S. Phys. Rev. Lett. **53** (1984) 834
32. Wiltzius, P., Carnell, D.S. Phys. Rev. Lett. **56** (1986) 61
33. Richter, D., Binder, K., Ewen, B., Stühn, B. J. Phys. Chem. **88** (1984) 6618
34. Sangani, A.S., Mo, G. Phys. Fluids **8** (1996) 1990
35. Hoogerbrugge, P.J., Koelman, J.M.V.A. Europhys. Lett. **19** (1992) 155
36. Koelman, J.M.V.A., Hoogerbrugge, P.J. Europhys. Lett. **21** (1993) 369
37. Español, P., Warren, P. Europhys. Lett. **30** (1995) 191
38. Español, P. Phys. Rev. E **52** (1995) 1734
39. Groot, R., Warren, P. J. Chem. Phys. **107** (1997) 4423
40. Español, P. Phys. Rev. E **57** (1998) 2930
41. Pagonabarraga, I., Hagen, M.J.H., Frenkel, D. Europhys. Lett. **42** (1998) 377

42. Gibson, J.B., Chen, K., Chynoweth, S. Int. J. Mod. Phys. C **10** (1999) 241
43. Besold, G., Vattulainen, I., Karttunen, M., Polson, J.M. Phys. Rev. E **62** (2000) R7611
44. Vattulainen, I., Karttunen, M., Besold, G., Polson, J.M. J. Chem. Phys. **116** (2002) 3967
45. Nikunen, P., Karttunen, M., Vattulainen, I. Comp. Phys. Comm. **153** (2003) 407
46. Shardlow, T. SIAM J. Sci. Comp. **24** (2003) 1267
47. Malevanets, A., Kapral, R. J. Chem. Phys. **110** (1999) 8605
48. Succi, S.: The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford University Press, Oxford (2001)
49. Ladd, A.J.C. Phys. Rev. Lett. **70** (1993) 1339
50. Ladd, A.J.C. J. Fluid Mech. **271** (1994) 285
51. Ladd, A.J.C. J. Fluid Mech. **271** (1994) 311
52. Ladd, A.J.C., Verberg, R. J. Stat. Phys. **104** (2001) 1191
53. Ladd, A.J.C., Gang, H., Zhu, J.X., Weitz, D.A. Phys. Rev. E **52** (1995) 6550
54. Lowe, C.P., Frenkel, D., Masters, A.J. J. Chem. Phys. **103** (1995) 1582
55. Hagen, M.H.J., Pagonabarraga, I., Lowe, C.P., Frenkel, D. Phys. Rev. Lett. **78** (1997) 3785
56. Hagen, M.H.J., Frenkel, D., Lowe, C.P. Physica A **272** (1999) 376
57. Heemels, M.W., Hagen, M.H.J., Lowe, C.P. J. Comput. Phys. **164** (2000) 48
58. Schneider, T., Stoll, E. Phys. Rev. B **17** (1978) 1302
59. Chandrasekhar, S. Rev. Mod. Phys. **15** (1943) 1
60. Risken, H.: The Fokker–Planck Equation. Springer–Verlag, Berlin (1984)
61. Dünweg, B.: Langevin methods. In Dünweg, B., Landau, D.P., Milchev, A.I., eds.: Computer Simulations of Surfaces and Interfaces. Kluwer, Dordrecht (2003)
62. Dünweg, B. J. Chem. Phys. **99** (1993) 6977
63. Müller-Plathe, F. J. Chem. Phys. **106** (1997) 6082
64. Müller-Plathe, F. Phys. Rev. E **59** (1999) 4894
65. Evans, D., Morriss, G.: Statistical Mechanics of Nonequilibrium Liquids. Academic Press, London (1990)
66. Soddemann, T., Dünweg, B., Kremer, K. Phys. Rev. E **68** (2003) 046702
67. Ihle, T., Kroll, D.M. Phys. Rev. E **67** (2003) 066705
68. Ihle, T., Kroll, D.M. Phys. Rev. E **67** (2003) 066706
69. Kikuchi, N., Pooley, C.M., Ryder, J.F., Yeomans, J.M. J. Chem. Phys. **119** (2003) 6388
70. Ihle, T., Tüzel, E., Kroll, D.M. Phys. Rev. E **70** (2004) 035701(R)
71. Ihle, T., Kroll, D.M. Comp. Phys. Comm. **129** (2000) 1
72. Swift, M.R., Orlandini, S.E., Osborn, W.R., Yeomans, J.M. Phys. Rev. E **54** (1996) 5041
73. Landau, L.D., Lifshitz, E.M.: Fluid mechanics. Addison-Wesley, Reading (1959)
74. Adhikari, R., Stratford, K., Cates, M.E., Wagner, A.J. Europhys. Lett. **71** (2005) 473
75. Ahlrichs, P., Dünweg, B. Int. J. Mod. Phys. C **9** (1998) 1429
76. Lobaskin, V., Dünweg, B. New J. Phys. **6** (2004) 54
77. Lobaskin, V., Dünweg, B., Holm, C. J. Physics: Cond. Matt. **16** (2004) S4063